
Advancing Local Clustering on Graphs via Compressive Sensing: Semi-supervised and Unsupervised Methods

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Local clustering aims to identify specific substructures within a large graph without
2 any additional structural information of the graph. These substructures are typically
3 small compared to the overall graph, enabling the problem to be approached by
4 finding a sparse solution to a linear system associated with the graph Laplacian. In
5 this work, we first propose a method for identifying specific local clusters when
6 very few labeled data are given, which we term semi-supervised local clustering.
7 We then extend this approach to the unsupervised setting when no prior information
8 on labels is available. The proposed methods involve randomly sampling the
9 graph, applying diffusion through local cluster extraction, then examining the
10 overlap among the results to find each cluster. We establish the co-membership
11 conditions for any pair of nodes, and rigorously prove the correctness of our
12 methods. Additionally, we conduct extensive experiments to demonstrate that the
13 proposed methods achieve state of the art results in the low-label rates regime.

14 1 Introduction

15 The ability to learn from data by uncovering its underlying patterns and grouping it into distinct
16 clusters based on latent similarities and differences is a central focus in machine learning. Over the
17 past few decades, traditional clustering problems have been extensively studied as an unsupervised
18 learning task, leading to the development of a wide range of foundational algorithms, such as k -means
19 clustering [MacQueen, 1967], density-based clustering [Ester et al., 1996], spectral clustering [Ng
20 et al., 2001, Zelnik-Manor and Perona, 2004], hierarchical clustering [Nielsen, 2016] and regularized
21 k -means [Kang et al., 2011]. These foundational methods have, in turn, inspired numerous variants
22 and adaptations tailored to specific data characteristics or application domains.

23 Researchers have also developed semi-supervised learning approach for clustering, which leverages
24 both labeled and unlabeled data in various learning tasks. One of the most commonly used methods
25 in graph-based semi-supervised learning is Laplace learning [Zhu et al., 2003]. Note that Laplacian
26 learning sometimes is also called Label propagation [Zhu and Ghahramani, 2002], which seeks a graph
27 harmonic function that extends the labels. Laplacian learning and its variants have been extensively
28 applied in semi-supervised learning tasks [Zhou et al., 2004a,b, 2005, Ando and Zhang, 2007, Kang
29 et al., 2014]. A key challenge with Laplacian learning type of algorithms is their poor performance in
30 the low-label rates regime. To address this, recent approaches have explored p -Laplacian learning
31 [El Alaoui et al., 2016, Flores et al., 2022], higher order Laplacian regularization [Zhou and Belkin,
32 2011], weighted nonlocal Laplacians [Shi et al., 2017], properly weighted Laplacian [Calder and
33 Slepcev, 2019], and Poisson learning [Calder et al., 2020].

Those aforementioned clustering algorithms are all global clustering algorithms, recovering all cluster structures simultaneously. However, real-world applications often require identifying only specific substructures within large, complex networks. For example, in a social network, an individual may only be interested in connecting with others who share similar interests, while disregarding the rest of individuals. In such cases, global clustering methods become inefficient, as they generate excessive redundant information rather than focusing on the relevant local patterns.

In this paper, we turn our focus to a more flexible clustering method called *local clustering* or *local cluster extraction*. Local clustering focus only on finding one target cluster which contains those nodes of interest to us, and disregard the non-interest or background nodes. Researchers have investigated in this direction over the recent decades such as [Lang and Rao, 2004, Andersen et al., 2006, Kloster and Gleich, 2014, Spielman and Teng, 2013, Andersen et al., 2016, Veldt et al., 2019, Fountoulakis et al., 2020]. More recently, inspired by the idea of compressive sensing, Lai and McKenzie [2020], Lai and Shen [2023], Shen et al. [2023] proposed a novel perspective for local clustering by finding the target cluster via solving a sparse solution to a linear system associated with the graph Laplacian matrix. Intuitively speaking, as local clustering only focus on finding “one cluster at a time”, it is flexible in practice and can be more efficient and effective than global clustering algorithms.

Besides such merits that local clustering algorithms possesses, one big downside of it is the necessity of having the initial nodes (we call them seeds) of interest as prior knowledge, and it also requires a good estimate of the target cluster size. The seeds information is sometimes very limited and even unavailable, which makes local clustering algorithm less popular. We propose a clustering method which requires very few seeds (semi-supervised case) or no seeds (unsupervised case), see Figure 1 and 2 for illustration. We provide a detailed discussion of the procedure, with analysis on the correctness of the proposed method. The main contributions of our work are as follows:

1. We propose a semi-supervised (with very few labeled data) and an unsupervised (no labeled data) local clustering methods which outperform the state-of-the-arts local and non-local clustering methods in most cases.
2. For theoretical considerations, we relax the assumption on the spectral norm of perturbation on the graph Laplacian and prove the correctness of our method in the semi-supervised case. We also establish the co-membership conditions for any pair of nodes, and then prove the correctness of our method in the unsupervised case.
3. Computationally the proposed method can also show benefits that our semi-supervised method can find all the clusters simultaneously which improves the “one cluster at a time” feature of local clustering algorithms in terms of efficiency. We provide extensive experiments with comparisons to show the effectiveness of our methods in the low-label rates regime.

The rest of the paper is organized as follows. In Section 2, we present step-by-step procedure of the proposed methods in both semi-supervised and unsupervised settings, and prove their correctness under certain assumptions. In section 5, we show the experimental results of the proposed methods and compare them with the state-of-the-art semi-supervised clustering algorithms on various benchmark datasets. Finally, in Section 6, we conclude the paper, discuss its limitations and societal impact.

2 Model Assumptions

For a graph $G = (V, E)$, we use V to denote the set of all nodes, and E to denote the set of all edges. Suppose G has k non-overlapping underlying clusters $C_1, C_2, \dots, C_k$, we use n_i to denote the size of C_i where $i = 1, 2, \dots, k$, and use n to denote the total size of graph G . Further, we use A to denote the adjacency matrix (possibly weighted but non-negative) of graph G , and use D to denote the diagonal matrix whose diagonal entries is the degree of the corresponding vertex. We make the following assumptions on the graph model.

Assumption 1 *The non-zero entries in the diagonal block is denser than the non-zero entries in the off-diagonal block (after permutation according to the cluster membership). More precisely, we assume the graph Laplacian satisfies $\|\Delta L\|_2 = \|L - L^{in}\|_2 = o(1)$ and $(\delta_n(L))^{\log n} = o(1)$ as the graph size $n \rightarrow \infty$. The definition of RIP constant δ_n is provided in Definition 1 in Appendix D.*

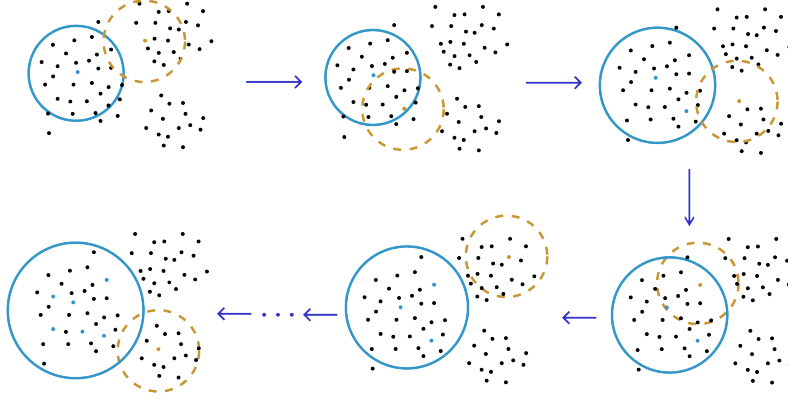


Figure 1: Illustration of semi-supervised local clustering (SSLC) for a single cluster. Each subplot indicates one iteration (Blue dots: seeds in Γ_1 . Brown dots: randomly sampled node in each iteration).

Assumption 2 The number of cluster $k = O(1) \geq 2$. The size of each cluster is not too small or too large, i.e., there exist $n_{\min} = \Theta(n)$ and $n_{\max} = \Theta(n)$ such that $n_{\min} \leq n_i \leq n_{\max}$ for any $i \geq 1$.

Assumption 3 For any distinct $i, j \in C_s$ with some $s \geq 1$, let $K_{i,j} \subset C_s$ be the set of nodes such that given any node in $K_{i,j}$ as the seed, LCE always finds node i and j into the same local cluster. We assume the cardinality of $K_{i,j}$ is at least $(1 - o(1)) \frac{n_{\min}^2}{n}$ for any distinct pair of nodes i, j .

Throughout the paper, all the notation $o(1)$ is with respect to the size of graph $n \rightarrow \infty$. Assumption 3 is essentially a type of homogeneity assumption on each cluster of the graph in order to guarantee the performance of LCE (see Algorithm 4) on a randomly chosen node.

One notable theoretical contribution in this paper is that we relax the assumption from $\|\Delta L\|_2 = o(n^{-1/2})$ (see Lemma 4 in Appendix C) [Shen et al., 2023] to $\|\Delta L\|_2 = o(1)$ (see Lemma 1 in Appendix A). We consider the local clustering tasks under two settings: semi-supervised case and unsupervised case.

3 Semi-supervised Local Clustering

For the semi-supervised case, we consider the following. Given a graph $G = (V, E)$ with underlying cluster $C_1, \dots, C_k$ such that $V = \cup_{i=1}^k C_i$, $C_i \cap C_j = \emptyset$ for $1 \leq i, j \leq k, i \neq j$.

1. Suppose a very small portion of seeds $\Gamma_1 \subset C_1$ is available, we would like to find all the nodes in the target cluster C_1 .
2. Suppose a very small portion of seeds $\Gamma_i \subset C_i$ is available for each $i = 1, \dots, k$, we would like to assign all nodes to their corresponding underlying clusters.

Similar to the issue of Laplacian learning type of algorithms, local clustering approach such as LCE (see Algorithm 4) becomes less effective in the low-label rates regime. Therefore one wants to extract more seeds from each cluster before applying LCE. We illustrate the idea in Figure 1. We assume our target cluster is C_1 , which is the cluster in the left of those three clusters illustrated in the first subplot of Figure 1.

The procedure of semi-supervised local clustering (SSLC) is summarized as follows:

1. Given a graph G , and initial seed(s) Γ_1 (indicated as the blue point in the first subplot), we first apply LCE to have a rough estimate $\tilde{C}_1$ (indicated by the solid blue circle) of C_1 .
2. Randomly sample a node v (the brown point in the first subplot) in G and apply LCE to get a rough cluster around v (indicated by the dashed brown circle). Then check the overlap between solid blue and dashed brown circle. If the overlap contains the majority of the nodes in the brown circle, then add v into Γ_1 , otherwise sample another node.

117 3. Continue this process and keep increasing the size of Γ_1 until a predetermined iteration
 118 number is reached (indicated in the process of from the second to the last subplots).

119 Note that this procedure can be applied when more than one cluster is of interest without increasing
 120 the number of iterations. The procedures for the single cluster case with Γ_1 and C_1 and multiple
 121 clusters case Γ_s and C_s are summarized in Algorithm 1 and 3 respectively. This method finds more
 122 seeds based on the initial seeds, and the user can flexibly determine the number of seeds wanted by
 123 increasing or decreasing the number of iterations. Theorem 1 essentially establishes the correctness
 124 of Algorithm 1 and 3.

Algorithm 1 Semi-supervised Local Clustering (SSLC) for a Single Cluster

Require: The adjacency matrix A of an underlying graph G , the initial seed(s) Γ_1 for the target cluster C_1 , the estimated size n_1 , the number of resampling iteration ℓ

Ensure: desired output cluster $C_1^\#$

```

1:  $\tilde{C}_1 \leftarrow \text{LCE}(A, n_1, \Gamma_1)$ 
2: for  $i = 1 : \ell$  do
3:    $v \leftarrow$  uniform randomly sampled seed node from  $G$ 
4:    $C^\# \leftarrow \text{LCE}(A, n_1, v)$ 
5:   if  $|\tilde{C}_1 \cap C^\#| > 0.5|C^\#|$  then
6:      $\Gamma_1 \leftarrow \Gamma_1 \cup \{v\}$ 
7:      $\tilde{C}_1 \leftarrow \text{LCE}(A, n_1, \Gamma_1)$ 
8:   end if
9: end for
10:  $C_1^\# \leftarrow \tilde{C}_1$ 

```

125 **Theorem 1** Suppose G satisfies Assumptions 1 - 3. Then when n (the size of G) gets large, for each
 126 $s = 1, \dots, k$, we have

$$\begin{cases} |\tilde{C}_s \cap C^\#| > (1 - o(1))|C^\#|, & \text{if } v \in C_s, \\ |\tilde{C}_s \cap C^\#| \leq o(|C^\#|), & \text{otherwise.} \end{cases}$$

127 Theorem 1 shows that whenever a node v being added to $\Gamma_1(\Gamma_s)$ based on the large overlap criterion,
 128 it satisfies $v \in C_1(C_s)$. This means that $\Gamma_1(\Gamma_s)$ only grow with the nodes within $C_1(C_s)$, which is
 129 the essential step to guarantee the correctness of Algorithm 1 and Algorithm 3.

130 4 Unsupervised Local Clustering

131 For unsupervised case, we assume the following: Given a graph $G = (V, E)$ with underlying
 132 cluster $C_1, C_2, \dots$, where $V = \cup_{i \geq 1} C_i$, $C_i \cap C_j = \emptyset$ for all $i \neq j$. Assume neither the prior
 133 information about seeds nor the number of cluster k is given, the goal is to assign all the nodes to
 134 their corresponding underlying clusters.

135 In this case, we randomly sample and build a local cluster from the sampled node every time. We
 136 check its overlap with the local cluster obtained from any newly sampled node. After a certain number
 137 of iterations, the nodes from the same underlying cluster are more likely to be clustered together. We
 138 illustrate the idea of unsupervised local clustering in Figure 2. The procedure of unsupervised local
 139 clustering (USLC) is summarized as follows:

- 140 1. Given a graph G , in each iteration, we randomly sample a node and find its local cluster via
 141 LCE (as shown in the top row of Figure 2).
- 142 2. Based on the found local cluster, build the co-membership matrix in such a way that it
 143 outputs 1 in the location (i, j) when both i, j are contained in that found cluster, and outputs
 144 0 otherwise.
- 145 3. Aggregate the clustering results from all iteration into a co-membership matrix M (roughly
 146 speaking, each entry $M_{i,j}$ represents the probability of a pair of nodes being clustered into
 147 the same local cluster). The values in each block of matrix M asymptotically converges to
 148 the same value, as the iteration number goes up (as shown in the bottom row of Figure 2).

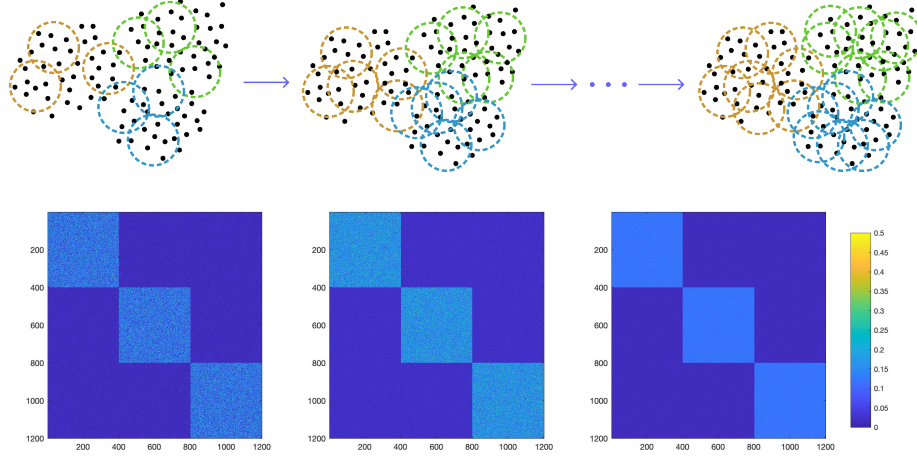


Figure 2: Illustration of USLC procedure. Top row: each dashed circle represents one iteration of LCE generated from a randomly sampled node, different colors indicate local clusters generated from nodes of different underlying clusters. Bottom row: Aggregated co-membership matrix.

- 149 4. Randomly sample a node i , and use a prescribed cutoff number δ to select all the nodes u
 150 such that $M_{i,j} > \delta$ and assign those j 's as one of the clusters $C^\#$.
- 151 5. Delete the subgraph generated by $C^\#$ from G . Then keep iterating until the prescribed
 152 maximum number of clusters is reached or the remaining size of graph is too small.

153 We summarize its procedure in Algorithm 2. We first show in Proposition 1 that $M_{i,j}$, the probability
 154 of a pair of nodes coming from the same underlying cluster, is significantly different from the
 155 probability of a pair of nodes coming from two different underlying clusters. Then we show in
 156 Theorem 2 the correctness of Algorithm 2. The detailed proofs are deferred to Appendix A.

Algorithm 2 Unsupervised Local Clustering (USLC)

Require: The adjacency matrix A of an underlying graph G , the number of resampling iteration ℓ

Ensure: The desired clusters $C_1^\#, C_2^\#, \dots$

```

1: initialize  $s \leftarrow 1$ 
2: while  $|G| < n_{\min}$  do
3:   initialize the comembership matrix  $M$  as zero matrix
4:   for  $i = 1 : \ell$  do
5:      $v \leftarrow$  uniform randomly sampled seed node from  $G$ 
6:      $C_i^\# \leftarrow \text{LCE}(A, n_{\min}, v)$ 
7:      $M \leftarrow M + \frac{1}{\ell} \cdot \text{comembership}(\mathbf{1}_{C_i^\#})$ 
8:   end for
9:   select a node  $v$  uniformly random such that  $M_{v,u} > \delta$  for some  $u \neq v$ , with some appropriate
      $\delta \in (0, 1)$ , then find all the  $j$  such that  $C_s^\# := \{j : M_{v,j} > \delta\}$ 
10:  Let  $G^{(s)}$  be the subgraph spanned by  $C_s^\#$ 
11:   $G \leftarrow G \setminus G^{(s)}$ 
12:   $s \leftarrow s + 1$ 
13: end while

```

157 It is worthwhile to mention that the value $n_{\min}$ can be a very rough lower bound, and Algorithm 2 is
 158 robust with respect to any reasonable choice of $n_{\min}$. Since the choice of δ for thresholding depends
 159 on $n_{\min}$, as long as we choose δ accordingly, the algorithm is robust. As shown in the proof Theorem
 160 2 in Appendix A, we can use $\delta = \frac{1}{2} \left(\frac{n_{\min}}{n} \right)^2$ as a general guideline in practice.

161 **Proposition 1** Suppose G satisfies Assumptions 1 - 3. Then, as n gets large, the co-membership
 162 matrix M obtained from Algorithm 2 has a clear block diagonal form (after permutation) as $L \rightarrow \infty$.

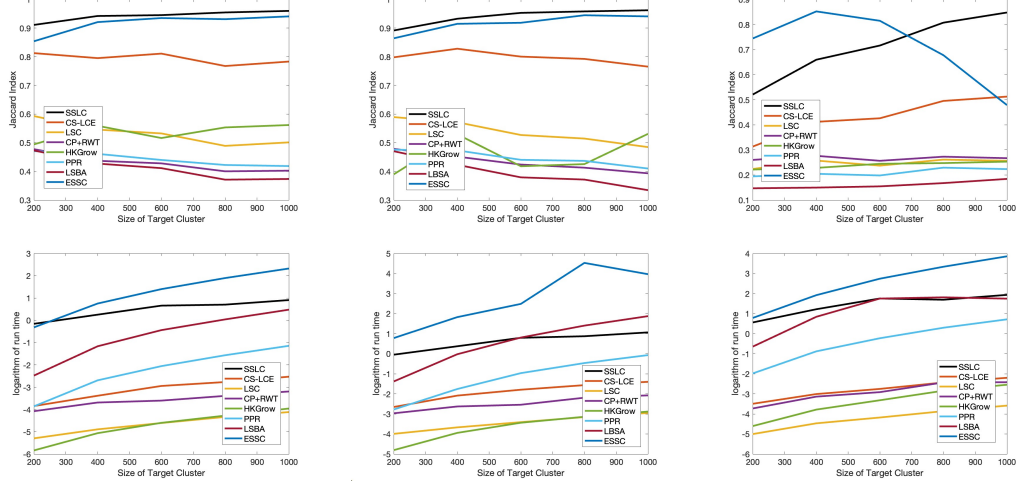


Figure 3: Plots of Jaccard index (over 100 trials) and logarithm of running time of SSLC for stochastic block model. (a) The first column shows experiment with three equal cluster size and single cluster C_1 extraction. (b) The middle column shows experiment with three equal cluster size and all clusters C_s extraction. (c) The right column shows experiment with unequal cluster sizes, focusing on the most dominant cluster.

163 *More precisely, the entries in M satisfy*

$$\begin{cases} M_{i,j} \geq (1 - o(1)) \frac{n_{\min}}{n}, & \text{for any } i = j, \\ M_{i,j} \geq (1 - o(1)) \frac{n_{\min}^2}{n^2}, & \text{for } i, j \in C_s, i \neq j, s \geq 1, \\ M_{i,j} \leq o\left(\frac{4n_{\min}^2(1-o(1))}{n^2}\right), & \text{otherwise.} \end{cases}$$

164 **Theorem 2** *Suppose the assumptions in Theorem 1 hold. For any $v \in C_s, s \geq 1$, there exists some*
 165 $\delta \in (0, 1)$ *and $C_s^\# = \{i : M_{v,i} > \delta\}$ such that $C_s^\# = C_s$. Consequently, Algorithm 2 assigns all*
 166 *nodes into their clusters correctly.*

167 It is worth of noting that Algorithms 1, 2 and 3 are applicable if there are outlier nodes, i.e., nodes
 168 which do not belong to any underlying clusters, presented in the graph. In such case, we extract all
 169 the clusters, and the remaining unclustered nodes are automatically classified as outlier nodes.

170 5 Experiments

171 We evaluate the performance of SSLC and USLC on both synthetic and real datasets, with a particular
 172 focus on clustering in the low-label rates regime (with label rates at most 1%). Additionally, we
 173 demonstrate the robustness of our methods by manually introducing outliers into the dataset and
 174 assessing their impact. All experiments are conducted on a local machine with 8-core Ryzen 7 7700X
 175 CPUs and 24 GB of RAM capacity. The runtime of SSLC/USLC is primarily determined by the LCE
 176 procedure, which is $O(nd_{\max} \log(n))$. In the regime $d_{\max} = O(\log n)$, this becomes $O(n \log^2 n)$.
 177 Therefore, if there are total k number of clusters, the total run time is $O(kn \log^2 n)$. Samples of demo
 178 code is available at https://anonymous.4open.science/r/Iterative_LCE-78C4.

179 **Synthetic Data** We first conduct two experiments for semi-supervised setting on stochastic block
 180 model (SBM) with different cluster sizes and different connection probabilities in Figure 3. The first
 181 cases (a) and (b) are symmetric SBM with three equal cluster size, the second case (c) is general
 182 SBM with three unequal clusters sizes. For the symmetric case, we consider both the single cluster
 183 extraction in (a) and all clusters extraction in (b). For the nonsymmetric case in (c), we only focus
 184 on extracting the most dominant cluster, i.e., the cluster with the largest connection probability. The
 185 parameters for generating data in both cases are presented in Appendix E. We compare against several
 186 other local clustering algorithms such as CS-LCE or LCE [Shen et al., 2023], LSC [Lai and Shen,

Table 1: Average accuracy and standard deviation on geometric data over 100 trials

Datasets	3 Lines	3 Circles	3 Moons
CP+RWT	82.1 (9.1)	96.1 (5.1)	85.4 (1.3)
LSC	89.0 (5.5)	96.2 (3.7)	85.3 (1.9)
LCE	92.4 (8.1)	97.6 (4.7)	96.8 (0.9)
SSLC	94.8 (7.2)	98.2 (4.1)	97.3 (1.2)

Table 2: Average accuracy and standard deviation on FashionMNIST over 30 trials

# Labels per class	0	1	2	3	4	5
Laplace	-	18.4 (7.3)	32.5 (8.2)	44.0 (8.6)	52.2 (6.2)	57.9 (6.7)
Random Walk	-	49.0 (4.4)	55.6 (3.8)	59.4 (3.0)	61.6 (2.5)	63.4 (2.5)
VolumeMBO	-	54.7 (5.2)	61.7 (4.4)	66.1 (3.3)	68.5 (2.8)	70.1 (2.8)
WNLL	-	44.6 (7.1)	59.1 (4.7)	64.7 (3.5)	67.4 (3.3)	70.0 (2.8)
p -Laplace	-	54.6 (4.0)	57.4 (3.8)	65.4 (2.8)	68.0 (2.9)	68.4 (0.5)
PoissonMBO	-	62.0 (5.7)	67.2 (4.8)	70.4 (2.9)	72.1 (2.5)	73.1 (2.7)
CutSSL	-	62.0 (5.7)	67.2 (4.8)	70.4 (2.9)	72.1 (2.5)	73.1 (2.7)
SSLC/USLC	61.6 (6.2)	65.8 (4.1)	69.1 (3.2)	74.3 (2.3)	77.2 (2.6)	78.7 (2.3)

2023], CP+RWT [Lai and McKenzie, 2020], HKGrow [Kloster and Gleich, 2014], PPR [Andersen et al., 2006], ESSC [Wilson et al., 2014], and LBSA [Shi et al., 2019].

For the experiments on SBM, we set the number of initial seeds to be 1. We observe that SSLC has a better Jaccard index compared to other methods in most cases. Moreover, for symmetric model, the Jaccard index in both single cluster extraction and all clusters extraction are similar, while the running time of all clusters extraction is more advantageous towards SSLC (see also Remark 2).

To further validate that our assumption $\|\Delta L\|_2 = \|L - L^{in}\|_2 = o(1)$, we conduct experiments on symmetric stochastic block model by fixing the number of nodes, number of clusters, intra-connection probability, while varying the inter-connection probability. The results are provided in Table 5. We observe that the $\|\Delta L\|_2$ decreases as n increases, which is as desired.

Our experimental results are also connected to the theoretical results in the SBM literature. Note that when the signal-to-noise ratio $\text{SNR} = \frac{(a-b)^2}{k(a+(k-1)b)} > 1$ (Kesten-Stigum threshold) in the SBM literature gives a theoretical bound for weak recovery of SBM in the sparse regime (with connectivity $\frac{a}{n}$ and $\frac{b}{n}$). For our exact recovery case (with connectivity $\frac{a \log n}{n}$ and $\frac{b \log n}{n}$), if we fix a, b, k and increase n , then $\|\Delta L\|$ decreases (illustrated in Table 5). In our experiments, we set $k = 3, p = \frac{a \log n}{n}, q = \frac{b \log n}{n}$ with $a = 6$ and $b = 1$, then we roughly satisfies the condition $\sqrt{a} - \sqrt{b} \approx \sqrt{k}$ for exact recovery, which is given in Theorem 14 and Remark 16 in Abbe [2018].

Besides SBM, we compare our algorithm against other compressive sensing based local clustering methods on geometric dataset consisting of different shapes (three lines, three circles, three moons). The 2D projection and more details of this dataset are provided in Appendix E.3. Note that spectral clustering often fails on these datasets when the noise level is large. However, compressive sensing based approaches work better. We use 1 label per class, the clustering results are shown in Table 1. We see that SSLC outperforms other compressive sensing based local clustering methods significantly.

Real Data For real datasets, we focus on clustering in the low-label rates regime on Fashion-MNIST [Xiao et al., 2017], CIFAR-10 [Krizhevsky et al., 2009], and Planetoid (Cora, CiteSeer, PubMed) [Yang et al., 2016]. To apply graph-based clustering algorithms on images, we first construct an auxiliary K-NN graph, and compute the pairwise distance from Gaussian kernel based on the Euclidean distance of the latent features with some scaling factors. Similar constructions have also appeared among others [Zelnik-Manor and Perona, 2004, Jacobs et al., 2018, Calder et al., 2020]. More detailed construction is provided in Appendix E.

We compare against other modern semi-supervised methods such as Laplace learning [Zhu et al., 2003], lazy random walks [Zhou and Scholkopf, 2004, Zhou et al., 2004a], weighted nonlocal

Table 3: Average accuracy and standard deviation on CIFAR-10 over 30 trials

# Labels per class	0	1	2	3	4	5
Laplace	-	10.4 (1.3)	11.0 (2.1)	11.6 (2.7)	12.9 (3.9)	14.1 (5.0)
Random Walk	-	36.4 (4.9)	42.0 (4.4)	45.1 (3.3)	47.5 (2.9)	49.0 (2.6)
VolumeMBO	-	38.0 (7.2)	46.4 (7.2)	50.1 (5.7)	53.3 (4.4)	55.3 (3.8)
WNLL	-	16.6 (5.2)	26.2 (6.8)	33.2 (7.0)	39.0 (6.2)	44.0 (5.5)
p -Laplace	-	26.0 (6.7)	35.0 (5.4)	42.1 (3.1)	48.1 (2.6)	49.7 (3.8)
PoissonMBO	-	41.8 (6.5)	50.2 (6.0)	53.5 (4.4)	56.5 (3.5)	57.9 (3.2)
CutSSL	-	41.8 (6.5)	50.2 (6.0)	53.5 (4.4)	56.5 (3.5)	57.9 (3.2)
SSLC/USLC	48.2 (6.1)	51.2 (5.2)	57.1 (4.9)	59.7 (4.7)	61.3 (3.4)	63.5 (2.7)

Table 4: Average accuracy and standard deviation of SSLC/USLC by adding 10% number of outliers

# Labels Per Class	0	1	2	3	4	5
FashionMNIST	55.6 (6.7)	60.4 (6.2)	65.6 (5.9)	69.9 (4.7)	73.5 (4.0)	74.8 (3.9)
CIFAR-10	45.1 (7.9)	47.3 (7.3)	51.6 (7.0)	55.3 (5.9)	58.1 (5.4)	60.1 (4.7)

laplacian (WNLL) [Shi et al., 2017], VolumeMBO [Jacobs et al., 2018], PoissonMBO [Calder et al., 2020], p -Laplace learning [Flores et al., 2022], and CutSSL [Holtz et al., 2024]. The results are summarized in Table 2 for FashionMNIST dataset, and Table 3 for CIFAR-10 dataset. Additional experimental results on Planetoid are included in Table 6 in Appendix F. Note that all the other methods can only handle the semi-supervised case, i.e., the number of label(s) per class is at least 1.

It is worthwhile to note that deep clustering method such as SimCLR [Chen et al., 2020] gives a strong baseline with 65% accuracy on FashionMNIST and 50% accuracy on CIFAR-10, which does not show clear advantage over our method, yet our method is much efficient and easy to implement.

To further demonstrate the robustness of our methods against outliers in the graph, we manually add some images consisting of random noise as outliers into datasets FashionMNIST and CIFAR-10. Specifically, we add the number of outliers equal to 10% of the original dataset size. Each outlier image is generated by setting its pixel values equal to the standard Gaussian random noise. For example, as shown in Figure 5, the last block consists entirely of outliers, exhibiting no community structure, while other blocks do. Such case is also called tight clustering in the literature [Tseng and Wong, 2005, Deng et al., 2024]. By comparing Table 4 with Table 2 and 3, it further confirms that the performance of our proposed methods remains largely unaffected in the presence of these outliers.

6 Conclusion and Discussion

Conclusion. We proposed a unified framework for extracting local clusters in both semi-supervised and unsupervised settings with theoretical guarantees. The methods require minimal label information in the semi-supervised case and no label information in the unsupervised case, achieving state-of-the-art results. Notably, the methods are very effective particularly in the low-label rates regime, and remain robust when outliers are presented. We point out two limitations of this work. First, our methods mostly work well for sparse graph. This is mainly due to sparse graph has clear clustering structure, meaning that it often have tightly connected clusters with fewer inter-cluster edges, making local algorithms efficient at detecting boundaries. Second, when the label rate is not low, our proposed method in the semi-supervised case do not exhibit advantages over other methods.

Broader Impact. While local clustering algorithms offer valuable insights for applications like community detection and recommendation systems, their potential misuse raises ethical concerns, particularly in contexts like surveillance or discriminatory profiling. By identifying tightly connected groups in social networks, these techniques can inadvertently reinforce biases (e.g., via homophily) or enable repression when deployed without transparency or consent. The societal impact hinges on intent—clustering can empower communities when used responsibly, but it risks harming marginalized groups if applied for unchecked monitoring or targeting. Practitioners should weigh considerations like purpose, bias mitigation, and user awareness to ensure ethical deployment, acknowledging that even mathematically neutral tools carry real-world consequences.

References

- Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.
- Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *IEEE Symposium on Foundations of Computer Science*, pages 475–486. IEEE, 2006.
- Reid Andersen, Shayan Oveis Gharan, Yuval Peres, and Luca Trevisan. Almost optimal local graph clustering using evolving sets. *Journal of the ACM (JACM)*, 63(2):1–31, 2016. doi: 10.1145/2856033. Article No. 15.
- R. K. Ando and T. Zhang. Learning on graph with laplacian regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 25–32. MIT Press, 2007.
- Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.
- J. Calder and D. Slepcev. Properly-weighted graph laplacian for semi-supervised learning. *Applied Mathematics & Optimization*, Dec 2019.
- Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepcev. Poisson learning: Graph based semi-supervised learning at very low label rates. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1306–1316. PMLR, 2020.
- Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.
- Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, May 2009.
- Jiayi Deng, Xiaodong Yang, Jun Yu, Jun Liu, Zhaiming Shen, Danyang Huang, and Huimin Cheng. Network tight community detection. In *Forty-first International Conference on Machine Learning*, 2024.
- David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of ℓ_p -based laplacian regularization in semi-supervised learning. In *Conference on Learning Theory (COLT)*, pages 879–906, 2016.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.
- Renzhong Feng, Aitong Huang, Ming-Jun Lai, and Zhaiming Shen. Reconstruction of sparse polynomials via quasi-orthogonal matching pursuit method. *Journal of Computational Mathematics*, 2021.
- Mauricio Flores, Jeff Calder, and Gilad Lerman. Analysis and algorithms for ℓ_p -based semi-supervised learning on graphs. *Applied and Computational Harmonic Analysis*, 60:77–122, 2022.
- Kimion Fountoulakis, Di Wang, and Shenghao Yang. p -norm flow diffusion for local graph clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3222–3232, 2020.

300 Matthew A. Herman and Thomas Strohmer. General deviants: An analysis of perturbations in
301 compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):342–349, 2010.

302 Chester Holtz, Pengwen Chen, Zhengchao Wan, Chung-Kuan Cheng, and Gal Mishne. Continuous
303 partitioning for graph-based semi-supervised learning. In *The Thirty-eighth Annual Conference on*
304 *Neural Information Processing Systems*, 2024. NeurIPS 2024.

305 M. Jacobs, E. Merkurjev, and S. Esedoglu. Auction dynamics: A volume constrained mbo scheme.
306 *Journal of Computational Physics*, 354:288–310, 2018.

307 Sung Ha Kang, Berta Sandberg, and Andy M Yip. A regularized k-means and multiphase scale
308 segmentation. *Inverse Problems and Imaging*, 5(2):407–429, 2011.

309 Sung Ha Kang, Behrang Shafei, and Gabriele Steidl. Supervised and transductive multi-class
310 segmentation using p-laplacians and rkhs methods. *Journal of Visual Communication and Image*
311 *Representation*, 25(5):1136–1148, 2014.

312 Kyle Kloster and David F. Gleich. Heat kernel based community detection. In *Proceedings of the*
313 *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages
314 1386–1395. ACM, 2014.

315 Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

316 Ming-Jun Lai and Daniel Mckenzie. Compressive sensing for cut improvement and local clustering.
317 *SIAM Journal on Mathematics of Data Science*, 2(2):368–395, 2020.

318 Ming-Jun Lai and Zhaiming Shen. A quasi-orthogonal matching pursuit algorithm for compressive
319 sensing. *arXiv preprint arXiv:2007.09534*, 2020.

320 Ming-Jun Lai and Zhaiming Shen. A compressed sensing based least squares approach to semi-
321 supervised local cluster extraction. *Journal of Scientific Computing*, 94(3):63, 2023.

322 Kevin Lang and Satish Rao. A flow-based method for improving the expansion or conductance
323 of graph cuts. In *Proceedings of the International Conference on Integer Programming and*
324 *Combinatorial Optimization (IPCO)*, pages 325–337, 2004.

325 Haifeng Li. Improved analysis of sp and cosamp under total perturbations. *EURASIP Journal on*
326 *Advances in Signal Processing*, 2016:1–6, 2016.

327 J. MacQueen. Classification and analysis of multivariate observations. In *Proceedings of the Fifth*
328 *Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

329 Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In
330 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 14, 2001.

331 Frank Nielsen. Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*, pages
332 195–211. 2016.

333 Yaghoub Rahimi, Sung Ha Kang, and Yifei Lou. A lifted ℓ_1 framework for sparse recovery. *Informa-*
334 *tion and Inference: A Journal of the IMA*, 13(1):iaad055, 2024.

335 Zhaiming Shen, Ming-Jun Lai, and Sheng Li. Graph-based semi-supervised local clustering with
336 few labeled nodes. In *Proceedings of the International Joint Conference on Artificial Intelligence*
337 *(IJCAI)*, pages 4190–4198, 2023.

338 Pan Shi, Kun He, David Bindel, and John E. Hopcroft. Locally-biased spectral approximation for
339 community detection. *Knowledge-Based Systems*, 164:459–472, 2019.

340 Z. Shi, S. Osher, and W. Zhu. Weighted nonlocal laplacian on interpolation from sparse data. *Journal*
341 *of Scientific Computing*, 73(2-3):1164–1177, 2017.

342 Daniel A. Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its
343 application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26,
344 2013. doi: 10.1137/110853996.

345 Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on*
346 *Information Theory*, 50:2231–2242, 2004.

347 George C Tseng and Wing H Wong. Tight clustering: a resampling-based approach for identifying
348 stable and tight patterns in data. *Biometrics*, 61(1):10–16, 2005.

349 Nate Veldt, Christine Klymko, and David F. Gleich. Flow-based local graph clustering with better
350 seed set inclusion. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*,
351 pages 378–386. SIAM, 2019.

352 Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

353 James D. Wilson, Simi Wang, Peter J. Mucha, Shankar Bhamidi, and Andrew B. Nobel. A testing
354 based extraction algorithm for identifying significant communities in networks. *The Annals of*
355 *Applied Statistics*, 8:1853–1891, 2014.

356 Han Xiao, Karim Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking
357 machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

358 Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with
359 graph embeddings. In *Proceedings of the 33rd International Conference on Machine Learning*,
360 volume 48 of *ICML’16*, pages 40–48. PMLR, 2016. Also introduces the Planetoid benchmark with
361 standardized splits for Cora, CiteSeer, and PubMed datasets.

362 Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural*
363 *Information Processing Systems (NeurIPS)*, volume 17, 2004.

364 D. Zhou and B. Scholkopf. Learning from labeled and unlabeled data using random walks. In *Joint*
365 *Pattern Recognition Symposium*, pages 237–244. Springer, 2004.

366 D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global
367 consistency. In *Advances in Neural Information Processing Systems*, pages 321–328. MIT Press,
368 2004a.

369 D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Scholkopf. Ranking on data manifolds. In
370 *Advances in Neural Information Processing Systems (NeurIPS)*, pages 169–176. MIT Press, 2004b.

371 D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph.
372 In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1036–1043.
373 ACM, 2005.

374 X. Zhou and M. Belkin. Semi-supervised learning by higher order regularization. In *Proceedings of*
375 *the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages
376 892–900, 2011.

377 Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propaga-
378 tion, 2002. ProQuest number: information to all users.

379 Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian
380 fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine*
381 *Learning (ICML-03)*, pages 912–919, 2003.

382 A Lemmas and deferred proofs

383 Let us introduce

$$\mathbf{x}^* := \arg \min_{\mathbf{x} \in \mathbb{R}^{|V|-|U|}} \{\|L_{V \setminus U}^{in} \mathbf{x} - \mathbf{y}^{in}\|_2 : \|\mathbf{x}\|_0 \leq \hat{n}_1 - |U|\}, \quad (1)$$

384 where $\mathbf{y}^{in} := -\sum_{i \in V \setminus U} \ell_i^{in} = L^{in} \mathbf{1}_{V \setminus U}$, and

$$\mathbf{x}^\# := \arg \min_{\mathbf{x} \in \mathbb{R}^{|V|-|U|}} \{\|L_{V \setminus U} \mathbf{x} - \mathbf{y}\|_2 : \|\mathbf{x}\|_0 \leq \hat{n}_1 - |U|\}, \quad (2)$$

385 where $\mathbf{y} := -\sum_{i \in V \setminus U} \ell_i = L \mathbf{1}_{V \setminus U}$. The solution $\mathbf{x}^*$ gives the underlying true node indices
386 associated with L^{in} , while $\mathbf{x}^\#$ gives the algorithmic output node indices associated with L .

387 We highlight the following Lemma 1, which is a crucial step in establishing the convergence result in
388 LCE. We relax the assumption from $\|\Delta L\|_2 = o(n^{-1/2})$ (see Lemma 4 in Appendix C) [Shen et al.,
389 2023] to $\|\Delta L\|_2 = o(1)$.

390 **Lemma 1** *Let $\Delta L := L - L^{in}$. Suppose $U \subset C_1$ and $0.1|C_1| < |U| < 0.9|C_1|$. Suppose further
391 that $\|\Delta L\|_2 = o(1)$ and $(\delta_{3(|C_1|-|U|)}(L))^{\log(n)} = o(1)$. Then*

$$\frac{\|\mathbf{x}^\# - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} = o(1). \quad (3)$$

392 Lemma 1 establishes the convergence result between $\mathbf{x}^\#$ and $\mathbf{x}^*$ when ΔL is small, which is the key
393 step for establishing the correctness of our algorithms in Theorem 1.

394 **Lemma 2** *Suppose G satisfies Assumptions 1 - 3. Let v be any sampled node with $v \in C_s$ for some
395 $s \geq 1$. Then, as n gets large, the cardinality of the set $S := \{i \in C_r : \text{comembership}(\mathbf{1}_{C^\#})_{v,i} =$
396 $\text{comembership}(\mathbf{1}_{C_s})_{v,i}\}$ satisfies*

$$\begin{cases} |S| \geq (1 - o(1))n_{\min}, & \text{if } r = s, \\ |S| \leq o(n_{\min}), & \text{otherwise.} \end{cases}$$

397 A.1 Proof of Lemma 1

398 *Proof.* [Proof of Lemma 1] First, since $\|L - L^{in}\|_2 = o(1)$, we have

$$\frac{\|\Delta \mathbf{y}\|_2}{\|\mathbf{y}\|_2} = \frac{\|\mathbf{y} - \mathbf{y}^{in}\|_2}{\|\mathbf{y}\|_2} = \frac{\|(L - L^{in})\mathbf{1}_{V \setminus U}\|_2}{\|L\mathbf{1}_{V \setminus U}\|_2} \leq \frac{o(1)\sqrt{n}}{\|L\mathbf{1}_U\|_2}.$$

399 By Assumption 2,, the cluster C_1 is on the same order as the size of the graph n , hence $U \subset C_1$ is
400 also on the same order as n . By Assumption 3, all the nodes asymptotically have the same degree,
401 therefore $\|L\mathbf{1}_U\|_2 = \Theta(\|\mathbf{1}_U\|_2) = \Theta(\sqrt{n})$. Hence

$$\frac{\|\mathbf{y} - \mathbf{y}^{in}\|_2}{\|\mathbf{y}\|_2} \leq \frac{o(1)\sqrt{n}}{\Theta(\sqrt{n})} = o(1).$$

402 Therefore, the quantity $\epsilon_{\mathbf{y}} = o(1)$ in Lemma 5.

403 With the same assumption $\|L - L^{in}\|_2 = o(1)$, it is not hard to see that, if the singular values
404 of L decays at a reasonable rate, then by applying the eigenvalue interlacing theorem, we have
405 $\epsilon_\Phi^s = o(1)$ as well (by letting $\Phi = L$ in Lemma 5 in Appendix D). With these, the second term on the
406 right-hand-side of the inequality in Lemma 5 satisfies

$$\tau \frac{\sqrt{1 + \delta_s}}{1 - \epsilon_\Phi^s} (\epsilon_\Phi^s + \epsilon_{\mathbf{y}}) \leq o(1).$$

407 Furthermore, since $\mathbf{x}^\#$ is the output of Algorithm 4 after $m = O(\log n)$ iteration, we have $\rho^m =$
408 $O((\delta_{3(|C_1|-|U|)}(L))^{\log n}) = o(1)$. Putting these together gives

$$\frac{\|\mathbf{x}^\# - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} = o(1)$$

409 as desired. $\square$

410 A.2 Proof of Theorem 1

411 *Proof.* [Proof of Theorem 1] For any $s \geq 1$, let us case on whether $v \in C_s$ or not. If $v \in C_s$, then by
 412 Lemma 1 and Theorem 4, we have

$$|C^\# \Delta(\tilde{C}_s \cap C^\#)| \leq |C^\# \Delta(C_s \cap C^\#)| + |(C_s \Delta \tilde{C}_s) \cap C^\#| \leq o(|C^\#|) + o(|C^\#|) = o(|C^\#|).$$

413 Therefore, $|\tilde{C}_s \cap C^\#| \geq |C^\#| - o(|C^\#|) = (1 - o(1))|C^\#|$.

414 If $v \in C_t, t \neq s$, then $|C_s \cap C^\#| \leq o(|C^\#|)$. We have

$$|C^\# \Delta(\tilde{C}_s \cap C^\#)| \geq |C^\# \Delta(C_s \cap C^\#)| - |(C_s \Delta \tilde{C}_s) \cap C^\#| \geq |C^\#| - o(|C^\#|) - o(|C^\#|) = |C^\#| - o(|C^\#|).$$

415 Therefore, $|\tilde{C}_s \cap C^\#| \leq |C^\#| - (|C^\#| - o(|C^\#|)) = o(|C^\#|)$.

416

□

417 A.3 Proof of Lemma 2

418 *Proof.* [Proof of Lemma 2] By Lemma 1 and Theorem 4, we have $|C^\# \Delta(C_s \cap C^\#)| \leq o(C^\#) =$
 419 $o(n_{\min})$. Therefore,

$$|C^\# \Delta C_s| = |C^\# \Delta(C_s \cap C^\#)| + |C_s \setminus C^\#| \leq o(n_{\min}) + n_s - n_{\min}.$$

420 Hence

$$\begin{aligned} |\{i \in C_s : \text{comembership}(\mathbf{1}_{C^\#})_{v,i} \neq \text{comembership}(\mathbf{1}_{C_s})_{v,i}\}| &= |(C^\# \Delta C_s) \cap C_s| \leq |C^\# \Delta C_s| \\ &\leq o(n_{\min}) + n_s - n_{\min}. \end{aligned}$$

421 So we conclude

$$\begin{aligned} |\{i \in C_s : \text{comembership}(\mathbf{1}_{C^\#})_{v,i} = \text{comembership}(\mathbf{1}_{C_s})_{v,i}\}| &\geq n_s - (o(n_{\min}) + n_s - n_{\min}) \\ &= n_{\min} - o(n_{\min}). \end{aligned}$$

422 As $C_s \cap C_t = \emptyset$, we have $|C^\# \cap C_t| \leq o(n_{\min})$. Therefore,

$$|\{i \in C_t : \text{comembership}(\mathbf{1}_{C^\#})_{v,i} = \text{comembership}(\mathbf{1}_{C_t})_{v,i}\}| \leq o(n_{\min}).$$

423

□

424 A.4 Proof of Proposition 1

425 *Proof.* [Proof of Proposition 1] By Lemma 2, we have the following estimates for the entries in the
 426 comembership matrix M asymptotically.

427 For any $i = j \in C_s$, some $s \geq 1$, the entries in M satisfies

$$M_{i,i} \geq \frac{n_s}{n} \cdot \frac{(1 - o(1))n_{\min}}{n_s} = (1 - o(1)) \frac{n_{\min}}{n}.$$

428 For any distinct $i, j \in C_s, s \geq 1$, by Assumption 3, the entries in M satisfies

$$M_{i,j} \geq \frac{1}{n} \cdot \frac{n_{\min}^2(1 - o(1))}{n} = (1 - o(1)) \frac{n_{\min}^2}{n^2}.$$

429 For any distinct i, j such that $i \in C_s, j \in C_t, s \neq t$, the entries in M satisfies

$$\begin{aligned} M_{i,j} &\leq \frac{n_s}{n} \cdot \frac{(1 - o(1))n_{\min}}{n_s} \cdot \frac{o(n_{\min})}{n - n_s} + \frac{n_t}{n - n_t} \cdot \frac{(1 - o(1))n_{\min}}{n_t} \cdot \frac{o(n_{\min})}{n} \\ &\leq \frac{2n_{\min}^2(1 - o(1)) \cdot o(1)}{n^2} + \frac{2n_{\min}^2(1 - o(1)) \cdot o(1)}{n^2} \\ &= o\left(\frac{4n_{\min}^2(1 - o(1))}{n^2}\right). \end{aligned}$$

430

□

431 A.5 Proof of Theorem 2

432 *Proof.* [Proof of Theorem 2] By direct computation, we can choose $\delta = \frac{1}{2}(\frac{n_{\min}}{n})^2$ such that

$$(1 - o(1))\frac{n_{\min}}{n} > \delta > o\left(\frac{4n_{\min}^2(1 - o(1))}{n^2}\right).$$

433 and

$$(1 - o(1))\frac{n_{\min}^2}{n^2} > \delta > o\left(\frac{4n_{\min}^2(1 - o(1))}{n^2}\right).$$

434 Hence, for any $i \in C_s$ satisfies $M_{v,i} > \delta$, and any $i \in C_t, t \neq s$, satisfies $M_{v,i} < \delta$. So we conclude
435 $C_s^\# = C_s$. $\square$

436 **Remark 1** Note that All the statements in Theorem 1 and 2, Lemma 1 and 2, and Proposition 1, are
437 asymptotic statements. The $o(\cdot)$ notation is with respect to $n \rightarrow \infty$.

438 B Additional Algorithms

Algorithm 3 Semi-supervised Local Clustering (SSLC) for Multiple Clusters

Require: The adjacency matrix A of an underlying graph G , the number of cluster k , the initial seed(s) Γ_s for each cluster, the estimated size n_s for each cluster, $s = 1, \dots, k$, the number of resampling iteration ℓ

Ensure: desired output clusters $C_1^\#, \dots, C_k^\#$

```

1: for  $s = 1 : k$  do
2:    $\tilde{C}_s \leftarrow \text{LCE}(A, n_s, \Gamma_s)$ 
3: end for
4: for  $i = 1 : \ell$  do
5:    $v \leftarrow$  uniform randomly sampled seed node from  $G$ 
6:    $C^\# \leftarrow \text{LCE}(A, \min\{n_s\}_{s \geq 1}, v)$ 
7:   if  $|\tilde{C}_s \cap C^\#| > 0.5|C^\#|$  for some  $s$  then
8:      $\Gamma_s \leftarrow \Gamma_s \cup \{v\}$ 
9:      $\tilde{C}_s \leftarrow \text{LCE}(A, n_s, \Gamma_s)$ 
10:  end if
11: end for
12: for  $s = 1 : k$  do
13:    $C_s^\# \leftarrow \tilde{C}_s$ 
14: end for
```

439 We use $C^\#$ to indicate the output of one of the LCE steps after sampling a random node v . Since
440 v can be from any cluster s , the notation $C^\#$ is independent of s . Due to Theorem 1, the condition
441 $|\tilde{C}_s \cap C^\#| > 0.5|C^\#|$ can only happen for one particular s . Therefore, Step 7 in Algorithm 3 will
442 only add each v to one of the clusters.

443 **Remark 2** One key advantage of Algorithm 3 is its ability to simultaneously identify all clusters,
444 while previous method such as LCE can only detect one cluster at a time. This characteristic provides
445 greater flexibility and makes Algorithm 3 significantly more efficient for practical applications,
446 particularly when the number of clusters is large. This advantage is also reflected in the first and
447 second columns of Figure 3.

448 C Review on Local Cluster Extraction (LCE)

449 The following result is central to our proposed sparse solution based local clustering method. We
450 omit its proof by referring to Von Luxburg [2007].

451 **Lemma 3** Let G be an undirected graph with non-negative weights. The multiplicity k of the eigen-
452 value zero of the graph Laplacian $L := I - D^{-1}A$ equals to the number of connected components
453 $C_1, C_2, \dots, C_k$ in G . Further, the indicator vectors $\mathbf{I}_{C_1}, \dots, \mathbf{I}_{C_k} \in \mathbb{R}^n$ on these components span
454 the kernel of L .

For the convenience of our discussion, let us introduce more notations. For a graph $G = (V, E)$ with certain underlying community structure, it is convenient to write $G = G^{in} \cup G^{out}$, where $G^{in} = (V, E^{in})$, $G^{out} = (V, E^{out})$. Here E^{in} is the set of all intra-connection edges within the same community (cluster), E^{out} is the remaining edges in E . Further, we use A^{in} and L^{in} to denote the adjacency matrix and Laplacian matrix associated with G^{in} respectively. In practice, we do not guarantee knowledge of the cluster to which each individual vertex belongs, meaning that A^{in} and L^{in} are not directly accessible. Instead, we only have access to A and L . A summary of the notations being used throughout this paper is included in Table 7 in Appendix G.

Let us first briefly introduce the idea Local Cluster Extraction (LCE), which applies the idea of compressive sensing (or sparse solution) technique to extract the target cluster in a semi-supervised manner. See also [Lai and McKenzie, 2020, Lai and Shen, 2023, Shen et al., 2023] for references.

Suppose the graph G consists of k connected equal-size components $C_1, \dots, C_k$, in other words, there is no edge connection between different clusters, i.e., $L = L^{in}$. For illustration purpose, let us permute the matrix according to the membership of each node, then L^{in} is in a block diagonal form i.e., all the off-diagonal blocks equal to zero:

$$L = L^{in} = \begin{pmatrix} L_{C_1}^{in} & & & \\ & L_{C_2}^{in} & & \\ & & \ddots & \\ & & & L_{C_k}^{in} \end{pmatrix}. \quad (4)$$

Suppose that the target cluster is C_1 . By Lemma 3, $\{\mathbf{1}_{C_1}, \dots, \mathbf{1}_{C_k}\}$ forms a basis of the kernel W_0 of L . Note that all the $\mathbf{1}_{C_i}$ have disjoint supports, so for $\mathbf{w} \in W_0$ and $\mathbf{w} \neq 0$, we have write

$$\mathbf{w} = \sum_{i=1}^k \alpha_i \mathbf{1}_{C_i} \quad (5)$$

with some $\alpha_i \neq 0$. If a prior knowledge is given that first node $v_1 \in C_1$, then the first cluster C_1 can be found by solving

$$\min \|\mathbf{w}\|_0 \quad \text{s.t.} \quad L^{in} \mathbf{w} = \mathbf{0} \quad \text{and} \quad w_1 = 1, \quad (6)$$

which gives the solution $\mathbf{w} = \mathbf{1}_{C_1} \in \mathbb{R}^n$ as desired. It is worthwhile to note that (6) is equivalent to

$$\min \|\mathbf{w}_{-1}\|_0 \quad \text{s.t.} \quad L_{-1}^{in} \mathbf{w}_{-1} = -\ell_1, \quad (7)$$

where L_{-1}^{in} is a submatrix of L^{in} with first column being removed, ℓ_1 is the first column of L^{in} . The solution to (7) is $\mathbf{w}_{-1} = \mathbf{1}_{C_1 \setminus v_1} \in \mathbb{R}^{n-1}$ which encodes the same index information as the solution to (6). The benefit of formulation (7) is that it can be solved by compressive sensing algorithms. Note that the sparse solution from solving (6) or (7) always gives the indices corresponding to the target cluster. Therefore, the permutation does not affect our analysis and clustering result. The above procedure is named CS-LCE or LCE in [Shen et al., 2023]. We summarize LCE as Algorithm 4 in Appendix C and provide brief analysis. We also briefly introduce the compressive sensing and sparse solution technique in Appendix D.

The following results are established in [Shen et al., 2023], and we refer the proofs to [Shen et al., 2023]. Let L^{in} be the matrix as in (4), then the true solution $\mathbf{x}^*$ which encodes the local cluster information is

$$\mathbf{x}^* := \arg \min_{\mathbf{x} \in \mathbb{R}^{|V|-|U|}} \{\|L_{V \setminus U}^{in} \mathbf{x} - \mathbf{y}^{in}\|_2 : \|\mathbf{x}\|_0 \leq \hat{n}_1 - |U|\} \quad (8)$$

where $\mathbf{y}^{in} = -\sum_{i \in V \setminus U} \ell_i^{in} = L^{in} \mathbf{1}_{V \setminus U}$.

Theorem 3 Suppose $U \subset C_1$ and $0.1|C_1| < |U| < 0.9|C_1|$. Then $\mathbf{x}^* = \mathbf{1}_{C_1 \setminus U} \in \mathbb{R}^{|V|-|U|}$ is the unique solution to (8).

Lemma 4 Let $\Delta L := L - L^{in}$. Suppose $U \subset C_1$ and $0.1|C_1| < |U| < 0.9|C_1|$. Suppose further that $\|\Delta L\|_2 = o(n^{-\frac{1}{2}})$ and $\delta_{3(|C_1|-|U|)}(L) = o(1)$. Then

$$\frac{\|\mathbf{x}^\# - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} = o(1). \quad (9)$$

491 **Theorem 4** Under the same assumption as Lemma 4. Then

$$\frac{|C_1 \triangle C_1^\#|}{|C_1|} \leq o(1). \quad (10)$$

492 We make use of Lemma 1 (improved version of Lemma 4) and Theorem 4 in our main proofs. The
493 LCE algorithm is summarized in Algorithm 4.

Algorithm 4 Compressive Sensing of Local Cluster Extraction (CS-LCE or LCE [Shen et al., 2023])

Require: Adjacency matrix A , the seed(s) set $\Gamma \subset C$ for some target cluster C , a known estimated size $\hat{n}_1 \approx |C_1|$, a fixed number of random walk depth $t \in \mathbb{Z}^+$, sparsity parameter $\gamma \in [0.1, 0.5]$, random walk threshold parameter $\epsilon \in (0, 1)$, rejection parameter $R \in [0.1, 0.9]$.

Ensure: the output target cluster $C^\#$

- 1: Compute $L = I - D^{-1}A$, $P = AD^{-1}$, $\mathbf{v}^0 = D\mathbf{1}_\Gamma$ and $\mathbf{v}^{(t)} = P^t\mathbf{v}^{(0)}$
- 2: Define $\Omega = \mathcal{L}_{(1+\epsilon)\hat{n}}(\mathbf{v}^{(t)})$
- 3: Let U be the set of column indices of $\gamma \cdot |\Omega|$ smallest components of the vector $|L_\Omega^\top| \cdot |L\mathbf{1}_\Omega|$.
- 4: Set $\mathbf{y} := -\sum_{i \in V \setminus U} \ell_i = L\mathbf{1}_{V \setminus U}$. Let $\mathbf{x}^\#$ be the solution to

$$\arg \min_{\mathbf{x} \in \mathbb{R}^{|V|-|U|}} \{ \|L_{V \setminus U}\mathbf{x} - \mathbf{y}\|_2 : \|\mathbf{x}\|_0 \leq \hat{n}_1 - |U| \} \quad (11)$$

obtained by using $O(\log n)$ iterations of *Subspace Pursuit* Dai and Milenkovic [2009].

- 5: Let $C^\# = \{i : \mathbf{x}_i^\# > R\} \cup U$
-

494 The notation $\mathcal{L}$ is defined as

$$\mathcal{L}_s(\mathbf{v}) := \{i \in [n] : v_i \text{ among } s \text{ largest-in-magnitude entries in } \mathbf{v}\}.$$

495 D Background on Compressive Sensing and Sparse Solution Technique

496 The concept of compressive sensing (also called compressed sensing or sparse sampling) emerged
497 from fundamental challenges in signal acquisition and efficient data compression. At its core, it
498 addresses the inverse problem of recovering a sparse (or compressible) signal from a small number of
499 noisy linear measurements:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_0 \quad s.t. \quad \|\Phi\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon, \quad (12)$$

500 where $\Phi \in \mathbb{R}^{m \times n}$ is called sensing matrix (usually underdetermined), $\mathbf{y} \in \mathbb{R}^m$ is called measurement
501 vector, and the “zero quasi-norm” $\|\cdot\|_0$ counts the number of nonzero components in a vector. Among
502 the key contributors, Donoho [Donoho, 2006] and Candès, Romberg and Tao [Candès et al., 2006]
503 are widely credited with being the first to explicitly introduce this concept and make it popular. Since
504 then, two families of approaches such as thresholding type of algorithms [Blumensath and Davies,
505 2009] and greedy type of algorithms [Tropp, 2004, Feng et al., 2021, Lai and Shen, 2020] have been
506 developed based on the idea of compressive sensing. One particular type of greedy algorithms that
507 has garnered our attention is the subspace pursuit [Dai and Milenkovic, 2009].

508 One of the reasons behind the remarkable usefulness of compressive sensing lies in its robustness
509 against errors, including both additive and multiplicative types. More precisely, suppose we know
510 $\mathbf{y} = \Phi\mathbf{x}^*$ where $\mathbf{y}$ is the exact measurement of the acquired signal and Φ is the exact measurement
511 of the sensing matrix. However, we may only be able to access to the noisy version $\tilde{\mathbf{y}} = \mathbf{y} + \Delta\mathbf{y}$
512 and $\tilde{\Phi} = \Phi + \Delta\Phi$. In such case, we can still approximate the solution $\mathbf{x}^*$ well from the noisy
513 measurements $\tilde{\mathbf{y}}$ and $\tilde{\Phi}$, as explained in the work of [Herman and Strohmer, 2010]. A unified
514 framework of lifted ℓ_1 form is explored in [Rahimi et al., 2024]. One crucial concept which is often
515 employed in the compressive sensing algorithm is called Restricted Isometry Property (RIP).

516 **Definition 1 (Restricted Isometry Property)** Let $0 < s < m$ be an integer, and sensing matrix
517 $\Phi \in \mathbb{R}^{m \times n}$. Suppose there exists a constant $\delta_s > 0$ such that

$$(1 - \delta_s)\|\mathbf{x}\|_2^2 \leq \|\Phi\mathbf{x}\|_2^2 \leq (1 + \delta_s)\|\mathbf{x}\|_2^2 \quad (13)$$

for all $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{x}\|_0 \leq s$. Then the matrix Φ is said to have the Restricted Isometry Property (RIP) of order s . The smallest constant $\delta_s(\Phi)$ which makes (13) hold is called the Restricted Isometry Constant (RIC) of Φ .

For Subspace Pursuit algorithm, we have the result in Lemma 5 [Li, 2016].

Lemma 5 Let $\mathbf{x}^*$, $\mathbf{y}$, $\tilde{\mathbf{y}}$, Φ , $\tilde{\Phi}$ be as defined above, and for any $t \in [n]$, let $\delta_s := \delta_s(\tilde{\Phi})$. Suppose that $\|\mathbf{x}^*\|_0 \leq s$. Define the following constants:

$$\epsilon_{\mathbf{y}} := \|\Delta \mathbf{y}\|_2 / \|\mathbf{y}\|_2 \quad \text{and} \quad \epsilon_{\Phi}^s := \|\Delta \Phi\|_2^{(s)} / \|\Phi\|_2^{(s)}$$

where $\|M\|_2^{(s)} := \max\{\|M_S\|_2 : S \subset [n], \#(S) = s\}$ for any matrix M . Define further:

$$\rho := \frac{\sqrt{2\delta_{3s}^2(1 + \delta_{3s}^2)}}{1 - \delta_{3s}^2} \quad \text{and} \quad \tau := \frac{(\sqrt{2} + 2)\delta_{3s}}{\sqrt{1 - \delta_{3s}^2}}(1 - \delta_{3s})(1 - \rho) + \frac{2\sqrt{2} + 1}{(1 - \delta_{3s})(1 - \rho)}.$$

Assume that $\delta_{3s} < 0.4859$ and let $\mathbf{x}^{(m)}$ be the output of Algorithm 4 after m iterations. Then:

$$\frac{\|\mathbf{x}^* - \mathbf{x}^{(m)}\|_2}{\|\mathbf{x}^*\|_2} \leq \rho^m + \tau \frac{\sqrt{1 + \delta_s}}{1 - \epsilon_{\Phi}^s} (\epsilon_{\Phi}^s + \epsilon_{\mathbf{y}}).$$

E Implementation Details

E.1 Constructing KNN Graphs

Let $\mathbf{x}_i \in \mathbb{R}^n$ be the vectorization of an image or the feature extracted from an image, we define the following affinity matrix of the K -NN auxiliary graph based on Gaussian kernel:

$$A_{ij} = \begin{cases} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / \sigma_i \sigma_j} & \text{if } \mathbf{x}_j \in NN(\mathbf{x}_i, K), \\ 0 & \text{otherwise.} \end{cases}$$

Note that similar construction has also appeared among others [Zelnik-Manor and Perona, 2004, Jacobs et al., 2018, Calder et al., 2020]. To construct high-quality graphs, we trained autoencoders to extract key features from the image data, we adopt the same parameters as [Calder et al., 2020] for training autoencoders to obtain these features.

The notation $NN(\mathbf{x}_i, K)$ indicates the set of K -nearest neighbours of $\mathbf{x}_i$, and $\sigma_i := \|\mathbf{x}_i - \mathbf{x}_i^{(r)}\|$ where $\mathbf{x}_i^{(r)}$ is the r -th closest point of $\mathbf{x}_i$. Note that the above A_{ij} is not necessary symmetric, so we consider $\tilde{A}_{ij} = A^T A$ for symmetrization. Alternatively, one may also want to consider $\tilde{A} = \max\{A_{ij}, A_{ji}\}$ or $\tilde{A} = (A_{ij} + A_{ji})/2$. We use $\tilde{A}$ as the input adjacency matrix for our algorithms. In our experiments, the local scaling parameters are chosen to be $K = 15$, $r = 10$ for all three real image datasets.

E.2 Parameters of Synthetic Data Generating and Algorithms

In all implementations where the LCE are applied, we sampled the seeds Γ_i uniformly from each C_i for all the implementations. We fix the rejection parameter $R = 0.1$, the random walk depth $t = 3$, random walk threshold parameter $\epsilon = 0.8$, and the removal set parameter $\gamma = 0.2$ for all experiments.

For synthetic data, the symmetric stochastic block model consists of three equal size clusters of $n_1 = 200, 400, 600, 800, 1000$ with connection probability $p = 5 \log(3n_1)/(3n_1)$ and $q = \log(3n_1)/(3n_1)$. The general stochastic block model consists of clusters' sizes as $\mathbf{n} = (n_1, 2n_1, 5n_1)$ where n_1 is chosen from 200, 400, 600, 800, 1000, and the connection probability has the matrix form

$$P = [P_{11}, P_{12}, P_{13}; P_{21}, P_{22}, P_{23}; P_{31}, P_{32}, P_{33}]$$

with $P_{11} = \log^2(8n_1)/(6n_1)$, $P_{22} = \log^2(8n_1)/(12n_1)$, $P_{33} = \log^2(8n_1)/(30n_1)$, $P_{12} = P_{21} = P_{13} = P_{31} = P_{23} = P_{32} = \log(8n_1)/(6n_1)$. In the implementation of SSLC on synthetic data, we choose the iteration number $L = 60$ in the symmetric case and $L = 90$ in the nonsymmetric case.

552 In the implementation of SSLC on FashionMNIST and CIFAR-10 (both with and w/o outliers), we
 553 choose the iteration number to be $L = 50k$ where k is the number of seeds, i.e., $k = 1, 2, 3, 4, 5$. In
 554 the implementation of USLC on FashionMNIST and CIFAR-10 (both with and w/o outliers), we
 555 choose the iteration number to be $L = 1000$, and we set $\delta = 0.01$ for FashionMNIST and $\delta = 0.005$
 556 for CIFAR-10.

557 E.3 Geometric Dataset

558 **Three Lines** We generate three parallel lines in the x - y plane defined by:

- 559 • Line 1: $y = 0$ with $x \in [0, 6]$
- 560 • Line 2: $y = 1$ with $x \in [0, 6]$
- 561 • Line 3: $y = 2$ with $x \in [0, 6]$

562 For each line, we sample **1,200 points** uniformly at random. The embedding into $\mathbb{R}^{100}$ is performed
 563 as:

- 564 1. **Zero-padding:** $(x, y) \mapsto (x, y, 0, \dots, 0) \in \mathbb{R}^{100}$
- 565 2. **Noise addition:** Each coordinate z_i is perturbed as $z_i \leftarrow z_i + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, 0.15)$

566 **Three Circles** We construct three concentric circles with:

- 567 • Circle 1: Radius $r = 1.0$ (500 points)
- 568 • Circle 2: Radius $r = 2.4$ (1,200 points)
- 569 • Circle 3: Radius $r = 3.8$ (1,900 points)

570 Points are sampled uniformly along each circle, totaling **3,600 points**. The $\mathbb{R}^{100}$ embedding follows
 571 the same zero-padding and noise injection procedure as above.

572 **Three Moons** Three semicircular clusters are generated with:

- 573 • Moon 1: Upper semicircle, radius 1.0, centered at $(0, 0)$ (1,200 points)
- 574 • Moon 2: Lower semicircle, radius 1.5, centered at $(1.5, 0.4)$ (1,200 points)
- 575 • Moon 3: Upper semicircle, radius 1.0, centered at $(3, 0)$ (1,200 points)

576 Each dataset uses identical embedding:

$$(x, y) \mapsto (x, y, 0, \dots, 0) + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, 0.15^2 \mathbf{I}_{100})$$

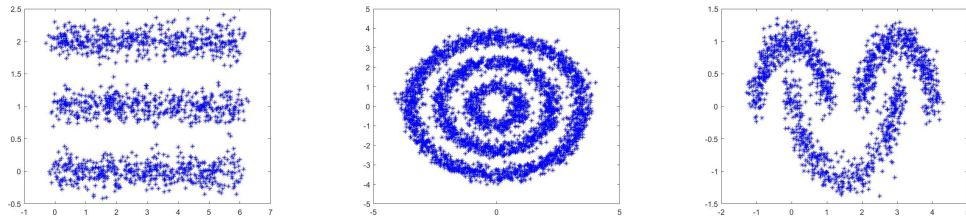


Figure 4: 2D Visualization of Geometric Dataset

Table 5: Jaccard Index of SSLC compared against the spectral norm of ΔL and SNR value on $\text{SSBM}(n, k, p, q)$ with $k = 3, p = 6 \log n/n, q = \log n/n$.

n	Jaccard Index (%)	Spectral norm of ΔL	SNR
100	84.59	0.4543	4.80
200	93.73	0.4238	5.52
400	96.04	0.3982	6.24
800	99.18	0.3725	6.96

577 F Additional Experimental Results

Table 6: Average accuracy and standard deviation on the largest connected subgraph over 100 trials

# Labels per class		0	1	3	5	10
Cora	Laplace (LP)	-	21.8 (14.3)	37.6 (12.3)	51.3 (11.9)	66.9 (6.8)
	Poisson	-	59.8 (7.9)	66.2 (5.8)	72.4 (2.1)	74.1 (1.8)
	PoissonMBO	-	59.9 (6.4)	69.1 (3.1)	72.4 (2.4)	74.3 (2.1)
	CutSSL	-	67.4 (3.4)	73.2 (3.1)	75.8 (2.1)	78.7 (1.1)
	SSLC/USLC	64.2 (5.7)	69.2 (4.0)	75.5 (3.4)	76.9 (2.6)	77.9 (1.3)
CiteSeer	Laplace (LP)	-	27.9 (10.4)	47.6 (8.1)	56.0 (5.9)	63.7 (3.5)
	Poisson	-	59.4 (5.4)	59.4 (5.4)	62.7 (4.2)	66.9 (1.8)
	PoissonMBO	-	47.7 (8.0)	55.7 (3.2)	61.0 (1.7)	63.1 (1.7)
	CutSSL	-	62.4 (4.6)	63.4 (7.2)	66.9 (1.4)	68.1 (1.3)
	SSLC/USLC	59.8 (5.7)	65.2 (5.3)	67.1 (4.7)	68.6 (2.9)	69.3 (1.6)
PubMed	Laplace (LP)	-	34.6 (8.8)	35.7 (8.2)	36.9 (8.1)	39.6 (9.1)
	Poisson	-	56.7 (12.8)	66.5 (6.6)	68.4 (5.9)	71.2 (3.4)
	PoissonMBO	-	56.9 (7.3)	67.9 (3.4)	69.6 (3.1)	71.4 (2.5)
	CutSSL	-	63.1 (4.7)	70.4 (3.1)	72.8 (2.9)	74.1 (1.4)
	SSLC/USLC	61.6 (6.2)	67.3 (5.1)	71.7 (3.9)	73.1 (2.9)	73.9 (2.2)

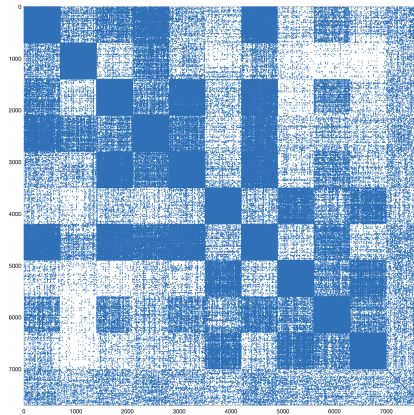


Figure 5: Affinity matrix of FashionMNIST after adding the outlier images (the last block consists of 10% outliers compared to the size of original dataset)

Table 7: Table of Notations

Symbols	Interpretations
G	general graph of interest
E	set of edges of graph G
V	set of nodes in G (size denoted by n)
C_s	each underlying true cluster
$C_s^\#$	each extracted cluster from algorithm
Γ_s	set of Seeds for each cluster
U	removal set from V in Algorithm 4
G^{in}	subgraph of G on V with edge set E^{in}
G^{out}	subgraph of G on V with edge set E^{out}
E^{in}	subset of E which consists only intra-connection edges
E^{out}	the complement of E^{in} within E
$A (A^{in})$	adjacency matrix of $G (G^{in})$
$L (L^{in})$	random walk Laplacian matrix of $G (G^{in})$
$L_C (L_C^{in})$	submatrix of $L (L^{in})$ with column indices $C \subset V$
$\ell_i (\ell_i^{in})$	i -th column of $L (L^{in})$
L_Ω^{in}	submatrix of L^{in} with column indices $\Omega \subset V$
$ M $	entrywised absolute value operation on matrix M
$\ M\ _2$	$\ \cdot\ _2$ norm of matrix M
$ \mathbf{v} $	entrywised absolute value operation on vector $\mathbf{v}$
$\ \mathbf{v}\ _2$	$\ \cdot\ _2$ norm of vector $\mathbf{v}$.
$\mathbf{1}_C$	indicator vector on subset $C \subset V$
Δ	set symmetric difference
$\mathcal{L}_s(\mathbf{v})$	$\{i \in [n] : v_i \text{ among } s \text{ largest-in-magnitude entries in } \mathbf{v}\}$

TAG-DS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The claims made in the abstract and introduction are accurately justified by experiments and theoretical results presented in the main text.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations and broader impact of this work is discussed in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We clearly state all assumptions in Section 2 and give complete proofs of all the results in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: To ensure the reproducibility, we give a complete description of the datasets usage, algorithms, and their hyperparameters in Section 5 and also in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and its usage for the experiments conducted in the paper are provided via an anonymous link in Section 5.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Experimental details are provided in Section 5 in the main article and also in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Standard deviations are reported in Section 5 in the main article and Section F in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide this information in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed the code of ethics and confirm our full compliance.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the societal impact in Section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite all the existing assets used in the paper and we follow all licensing.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code and documentation for the experiments are provided in an anonymous link in Section 5.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our work does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- 888 • We recognize that the procedures for this may vary significantly between institutions
889 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
890 guidelines for their institution.
891 • For initial submissions, do not include any information that would break anonymity (if
892 applicable), such as the institution conducting the review.

893 **16. Declaration of LLM usage**

894 Question: Does the paper describe the usage of LLMs if it is an important, original, or
895 non-standard component of the core methods in this research? Note that if the LLM is used
896 only for writing, editing, or formatting purposes and does not impact the core methodology,
897 scientific rigorousness, or originality of the research, declaration is not required.

898 Answer: [NA]

899 Justification: This research does not involve LLMs as any important, original, or non-
900 standard components.

901 Guidelines:

- 902 • The answer NA means that the core method development in this research does not
903 involve LLMs as any important, original, or non-standard components.
904 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
905 for what should or should not be described.