# Adversarial Graph Embedding for Ensemble Clustering

**Zhiqiang Tao**[1], **Hongfu Liu**[2], **Jun Li**[3], **Zhaowen Wang**[4] and **Yun Fu**[1,5]
[1]Department of Electrical and Computer Engineering, Northeastern University, Boston, MA
[2]Michtom School of Computer Science, Brandeis University, Waltham, MA
[3]Institute of Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA
[4]Adobe Research, Adobe Systems Incorporated, San Jose, CA
[5]Khoury College of Computer and Information Sciences, Northeastern University, Boston, MA
{zqtao, yunfu}@ece.neu.edu, hongfuliu@brandeis.edu, junli@mit.edu, zhawang@adobe.com

## Abstract

Ensemble clustering generally integrates basic partitions into a consensus one through a graph partitioning method, which, however, has two limitations: 1) it neglects to reuse original features; 2) obtaining consensus partition with learnable graph representations is still under-explored. In this paper, we propose a novel Adversarial Graph Auto-Encoders (AGAE) model to incorporate ensemble clustering into a deep graph embedding process. Specifically, graph convolutional network is adopted as probabilistic encoder to jointly integrate the information from feature content and consensus graph, and a simple inner product layer is used as decoder to reconstruct graph with the encoded latent variables (*i.e.*, embedding representations). Moreover, we develop an adversarial regularizer to guide the network training with an adaptive partition-dependent prior. Experiments on eight real-world datasets are presented to show the effectiveness of AGAE over several state-of-the-art deep embedding and ensemble clustering methods.

## 1 Introduction

Ensemble clustering (EC) [Strehl and Ghosh, 2003; Fred and Jain, 2005] takes as input a set of basic partitions (BPs) and targets to deliver a robust clustering result by integrating all the BPs into a consensus one. One representative and effective way to achieve consensus is based on the tool of co-association matrix [Fred and Jain, 2005], which in essence encodes the pairwise similarity between data points upon the categorical data (*i.e.*, BPs), and builds a tight link to the graph partitioning problem. In this paper, we call the graph defined by co-association matrix as *consensus graph*. Previous works [Fred and Jain, 2005; Luo *et al.*, 2011; Lancichinetti and Fortunato, 2012] show that the consensus graph is able to capture various cluster structures by exploiting the diversity among basic partitions.

However, although EC methods on consensus graph have achieved appealing performance, there still exits two limitations. First, as consensus graph is only built with categorical BPs, it neglects to reuse the rich information inside original features, which could badly degrade the clustering performance on some cases [Tao *et al.*, 2017a; Domeniconi and Al-Razgan, 2009]. Second, existing methods mainly conduct graph partitioning algorithm on a consensus graph directly, *e.g.*, spectral graph partitioning [Liu *et al.*, 2015], yet without considering to learn discriminative and low-dimensional graph representations. Nevertheless, recent research efforts [Xie *et al.*, 2016; Yang *et al.*, 2017] have shown that learning an *easy-partitioning* embedding space is quite crucial for data cluster analysis.

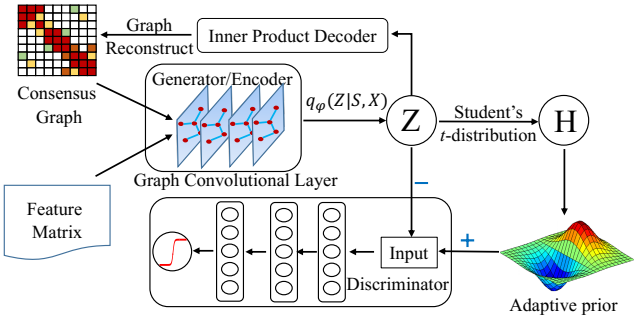



Figure 1: Illustration of the proposed AGAE. Adversarial training is adopted as a regularizer to guide the learning of embedding representations $\mathbf{Z}$, where we treat $\mathbf{Z}$ sampled from the encoded distribution $q_\phi$ as *fake* samples and $\mathbf{Z}$ from our adaptive partition-dependent prior as the *real* ones.

To address the above challenges, we propose a novel Adversarial Graph Auto-Encoders (AGAE) model in this study (see Fig. 1), where graph convolutional network (GCN) [Kipf and Welling, 2017] is used as a probabilistic encoder to model the posterior distribution over latent variables (*i.e.*, embedding representations), and a simple inner product layer is employed as decoder to reconstruct graph from the encoded distribution. By leveraging GCN, our model jointly encodes the consensus graph and original features, which not only exploits rich information from feature content, but also inherits a good cluster structure from ensemble clustering. Moreover, we develop an adversarial regularizer, where the probabilistic encoder plays the role of *generator* and a multi-layer neural network works as *discriminator*. By this means, an adaptive partition-dependent prior is imposed on the embedding representations to explicitly involve the clustering task.

The proposed AGAE model is built on the top of variational graph auto-encoders [Kipf and Welling, 2016] and adversarial learning [Makhzani *et al.*, 2016; Goodfellow *et al.*, 2014]. Some recent works [Dai *et al.*, 2018; Wang *et al.*, 2018; Pan *et al.*, 2018] also learn graph representations with adversarial training, which has shown promising performance in several tasks such as link prediction and node classification/clustering. However, they all focus on network embedding tasks and require input as graph-structured data. Different from these methods, we introduce AGAE to solve ensemble clustering (for generic data without real graph structure) in a learnable graph embedding way, and specifically design our model for the clustering purpose.

Experimental results on eight real-world datasets demonstrate the effectiveness of our approach for a clustering task, compared with several state-of-the-art deep embedding and ensemble clustering methods. Extensive model discussions are also provided to explore the impact of different regularizing and graph construction methods to our network. The contributions of this work are highlighted as follows.

- We propose a novel Adversarial Graph Auto-Encoders (AGAE) model, which develops an adversarial regularizer to guide the GCN based probabilistic encoder with an adaptive partition-dependent prior.
- We introduce an alternative objective function for training the deep graph embedding network, *i.e.*, by consensus graph reconstruction.
- The proposed AGAE encapsulates ensemble clustering in a learnable graph embedding process, which seamlessly integrates consensus graph and feature content.

# 2 Methodology

## 2.1 Consensus Graph

Ensemble Clustering (EC) integrates multiple "weak" basic partitions for a high-quality and robust clustering result. Given a set of $N$ data points $\mathcal{X} = \{x_1, \dots, x_N\}$ from $K$ clusters $\mathcal{C} = \{C_1, \dots, C_K\}$, and $M$ input basic partitions (BPs) $\Pi = \{\pi_1, \dots, \pi_M\}$, the goal of ensemble clustering is to find a consensus partition that agrees with all the BPs as much as possible. Each BP is obtained by running existing clustering algorithm and denoted as a cluster label set $\pi_r = \{\pi_r(x_1), \dots, \pi_r(x_N)\}$, where $1 \le \pi_r(x_i) \le K_r$, $1 \le i \le N$ and $1 \le r \le M$. $K_r$ is the cluster number of the $r^{th}$ BP, which is usually set to be different from the true cluster number $K$ to ensure the diversity among basic partitions [Fred and Jain, 2005; Wu *et al.*, 2015].

Ensemble clustering has a close connection to graph-based methods, among which co-association matrix [Fred and Jain, 2005] is an important tool to organize $\mathcal{X}$ as graph-structured data, upon the category information given by BPs. Specifically, the co-association matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ is defined as

$$\mathbf{S}_{i,j} = \frac{1}{M} \sum_{r=1}^{M} \delta(\pi_r(x_i), \pi_r(x_j)), \tag{1}$$

where $x_i, x_j \in \mathcal{X}$ and $\delta(a, b)$ is 1 if $a = b$; 0 otherwise. According to Eq. (1), $\mathbf{S}$ is in essence a pair-wise affinity matrix, which represents the probability of two data points assigned to the same cluster. However, $\mathbf{S}$ may have some noises and outliers due to the disagreement between different BPs, which may mislead the graph embedding process. To alleviate this problem, we construct the consensus graph by

$$\mathbf{S} = \text{sign}([\mathbf{S} - \tau]_+) \odot \mathbf{S}, \tag{2}$$

where $\odot$ denotes the element-wise Hadamard product. Eq. (2) is a simple thresholding operator parameterized by $\tau$ as $\mathbf{S}_{ij} = \mathbf{S}_{ij}$ if $\mathbf{S}_{ij} > \tau$; $\mathbf{S}_{ij} = 0$ otherwise. By this means, we could remove the low-confidence edges to some extent while holding the high-confidence ones. Using this simple trick, we build a sparse and robust consensus graph as the input for our AGAE model. It is worth noting that, due to the robustness nature of ensemble clustering, the proposed AGAE is quite insensitive to the parameter $\tau$ within a wide range, as will be shown in the experiment.

## 2.2 Graph Convolutional Network (GCN)

Graph convolutional network (GCN) extends the convolutional operation from low-dimensional regular grids to high-dimensional graph-structured data. In our model, we take GCN as a basic building block and focus on spectral graph convolution networks [Bruna *et al.*, 2013; Henaff *et al.*, 2015; Defferrard *et al.*, 2016; Kipf and Welling, 2017]. Let $\mathcal{G}$ be an undirected graph of $N$ nodes (*i.e.*, corresponding to $\mathcal{X}$) represented by $\mathbf{S}$ in Eq. (2) and $\mathbf{X} \in \mathbb{R}^{N \times d}$ be its feature matrix. Following [Kipf and Welling, 2017], the spectral graph convolution on $\mathcal{G}$ is defined by

$$\mathbf{Z} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{S}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{W}, \tag{3}$$

where $\tilde{\mathbf{S}} = \mathbf{I}_N + \mathbf{S}$, $\tilde{\mathbf{D}} = \tilde{\mathbf{S}} \cdot \mathbf{1}$, $\mathbf{W} \in \mathbb{R}^{d \times m}$ is the filter parameter matrix and $\mathbf{Z} \in \mathbb{R}^{N \times m}$ is the convolution result. $\mathbf{I}_N$ denotes the identity matrix and $\mathbf{1}$ is the vector of ones with compatible size. Upon Eq. (3), a multi-layer GCN model could be defined with the layer-wise propagation rule as

$$\mathbf{Z}^{(l+1)} = g(\mathbf{Z}^{(l)}, \mathbf{S}) = \xi(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{S}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{Z}^{(l)} \mathbf{W}^{(l)}), \tag{4}$$

where $g(\mathbf{Z}^{(l)}, \mathbf{S})$ refers to the graph convolution network, $\xi(\cdot)$ represents an activation function, such as the ReLU or sigmoid function, and $\mathbf{W}^{(l)} \in \mathbb{R}^{m_{l-1} \times m_l}$ ($m_0 = d$) is the $l^{th}$ layer's filter parameters matrix. $\mathbf{Z}^{(l)}$ denotes the hidden representation given by the $l^{th}$ GCN layer, while $\mathbf{Z}^{(0)}$ indicates the input feature matrix $\mathbf{X}$.

## 2.3 Adversarial Graph Auto-Encoders

The proposed Adversarial Graph Auto-Encoders (AGAE) model could be formulated by

$$\begin{aligned} \min_{\phi, \mathbf{H}} \max_{D} \; & \underbrace{\mathrm{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S})}[-\log p(\mathbf{S}|\mathbf{Z})]}_{\text{consensus graph reconstruction}} + \\ & \underbrace{\mathrm{E}_{p(\mathbf{Z}|\mathbf{H})}[\log D(\mathbf{Z})] + \mathrm{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S})}[\log(1 - D(\mathbf{Z}))]}_{\text{adversarial regularizer}}, \end{aligned} \tag{5}$$

where $q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{S})$ represents the GCN based probabilistic encoder network parameterized by $\phi$, and $D$ denotes a multi-layer neural network parameterized by $\psi$, *i.e.*, $D = f(\mathbf{z}; \psi)$ :

$\mathbb{R}^{m_2} \rightarrow \mathbb{R}^1$. During the adversarial training, $q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S})$ is used as the generator, and $D$ acts as the discriminator that tries to distinguish between *true samples* from the given prior and *fake samples* generated by the encoder. In Eq. (5), $p(\mathbf{Z}|\mathbf{H})$ denotes our adaptive prior and $\mathbf{H}$ is a soft partition matrix that represents the cluster assignment.

Jointly considering Eq. (1) and Eq. (4), our model solves ensemble clustering via consensus graph embedding, which learns the data distribution in a low-dimensional manifold to capture the discriminative information underling graph structure (*i.e.*, cluster structure). By this means, AGAE exhibits two benefits: 1) it jointly utilizes feature and partition information via a learnable network; 2) the low-dimensional embedding representations can facilitate the clustering task.

**GCN Based Auto-Encoders**

The probabilistic encoder of AGAE is modeled by a multivariate Gaussian distribution as

$$q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S}) = \prod_{i=1}^{N} q_\phi(\mathbf{Z}_i|\mathbf{X},\mathbf{S}), \tag{6}$$

where $q_\phi(\mathbf{Z}_j|\mathbf{X},\mathbf{S}) = \mathcal{N}(\mathbf{Z}_i|\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2))$, $\mathbf{Z}_i$ denotes the $i^{th}$ row in $\mathbf{Z}$. We represent $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ as the matrix of mean vectors and standard deviations, parameterized by GCN as

$$\begin{aligned} \boldsymbol{\mu} &= g_\mu(\mathbf{X},\mathbf{S}) = \mathbf{L_S}\xi(\mathbf{L_S}\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}_\mu^{(2)}, \\ \log \boldsymbol{\sigma} &= g_\sigma(\mathbf{X},\mathbf{S}) = \mathbf{L_S}\xi(\mathbf{L_S}\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}_\sigma^{(2)}, \end{aligned} \tag{7}$$

where $g_\mu$ and $g_\sigma$ represent a two-layer GCN for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively, $\mathbf{L_S} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{S}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ and $\xi(\cdot)$ is set to be the Relu function. In Eq. (7), $g_\mu$ and $g_\sigma$ share the weight of first layer and omit the activation function in the top layer for modeling the data distribution. Particularly, our encoder network is parameterized by $\phi$, *i.e.*, $\phi = \{\mathbf{W}^{(1)} \in \mathbb{R}^{d\times m_1}, \mathbf{W}_\mu^{(2)} \in \mathbb{R}^{m_1\times m_2}, \mathbf{W}_\sigma^{(2)} \in \mathbb{R}^{m_1\times m_2}\}$. From Eq. (6-7), the *reparameterization trick* [Kingma and Welling, 2014] is used to obtain an explicit expression for $\mathbf{Z}$ as

$$\mathbf{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \tag{8}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{N\times m_2}$ is an auxiliary variable with $\boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \mathbf{I}_{m_2})$. Eq. (8) makes the Monte Carlo estimate of the expectation of $q_\phi(\cdot)$ differentiable w.r.t $\phi$, which enables training AGAE with stochastic gradient descent (SGD) methods.

Following [Kipf and Welling, 2016], we simply define the decoder network with an inner product layer as

$$p(\mathbf{S}|\mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{N} s(\mathbf{Z}_i\mathbf{Z}_j^{\mathrm{T}}), \tag{9}$$

where $s(\cdot)$ denotes the sigmoid function. By using Eq. (6-9), we finally give the reconstruction loss $\mathcal{L}_{\mathrm{AE}}$ by

$$\begin{aligned} \mathcal{L}_{\mathrm{AE}} &= \mathrm{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S})}[-\log p(\mathbf{S}|\mathbf{Z})] \\ &= -\sum_{i,j=1}^{N} [y_{ij}\log\hat{y}_{ij} + (1-y_{ij})\log(1-\hat{y}_{ij})], \end{aligned} \tag{10}$$

where $y_{ij} = s(\mathbf{S}_{ij})$ and $\hat{y}_{ij} = s(\mathbf{Z}_i\mathbf{Z}_j^{\mathrm{T}})$.

***Algorithm 1.*** *Training of Adversarial Graph Auto-Encoder*

**Input:** Dataset $\mathbf{X}$ and consensus graph $\mathbf{S}$.
**Initial:** Initialize $\phi$ and $\psi$
**while** *not converged* **do** # *pre-training*
  Sample $\tilde{\mathbf{Z}}$ from $\mathcal{N}(\mathbf{0},\mathbf{I})$
  Update $\phi$ by minimizing $\mathcal{L}_{AE}$ in Eq. (10) with fixed $\psi$;
  Update $\psi$ by minimizing $\mathcal{L}_D$ in Eq. (14) with fixed $\phi$;
  Update $\phi$ by minimizing $\mathcal{L}_G$ in Eq. (13) with fixed $\psi$;
**end while**
Compute $\mathbf{Z}$ by Eq. (8);
Update $\theta$ by running K-means with $\mathbf{Z}$;
Calculate $\mathbf{H}$ by Eq. (12);
Update $p(\mathbf{Z}|\mathbf{H})$ by fitting GMM with $\mathbf{Z}$ and $\mathbf{H}$;
**while** *not converged* **do** # *fine-tunning*
  Sample $\tilde{\mathbf{Z}}$ from $p(\mathbf{Z}|\mathbf{H})$;
  Update $\{\phi,\psi\}$ by repeating Step 3-5;
  Update $p(\mathbf{Z}|\mathbf{H})$ by repeating Step 7-10;
**end while**
**Output:** $\phi$, $\psi$ and partition $\mathbf{H}$.

**Adaptive Partition-Dependent Prior**

Recent deep generative models [Kingma and Welling, 2014; Makhzani *et al.*, 2016] mainly employ a fixed isotropic Gaussian prior over the latent variables, which lacks an explicit guidance from the clustering task. Hence, we formulate our partition-dependent prior as a mixture of Gaussians by

$$p(\mathbf{Z}|\mathbf{H}) = \sum_{c=1}^{K} p(\mathbf{Z}|c)p(c|\mathbf{H}), \tag{11}$$

where $p(\mathbf{Z}|c) = \mathcal{N}(\mu_c, \sigma_c^2\mathbf{I})$, $p(c|\mathbf{H}) = \mathrm{Cat}(c|\pi)$ represents the categorical distribution, and $\pi$ is directly determined by $\mathbf{H}$. Inspired by [Xie *et al.*, 2016], the soft partition distribution $\mathbf{H}$ is defined by Student's $t$-distribution as following [v. d. Maaten and Hinton, 2008], which is given by

$$\mathbf{H}_{ik} = \frac{(1+\|\mathbf{Z}_i - \boldsymbol{\theta}_k\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K}(1+\|\mathbf{Z}_i - \boldsymbol{\theta}_{k'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{12}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{K\times m_2}$ denotes the cluster centers, $\alpha$ decides the degrees of the freedom of Student's $t$-distribution, and $1 \le i \le N$, $1 \le k \le K$. In this paper, we set $\alpha = 1$ as the default setting. According to Eq. (12), $\mathbf{H}_{ik}$ represents the probability of the $i^{th}$ data point belonging to the $k^{th}$ cluster.

By using Eq. (11), the proposed AGAE tries to disentangle the embedding space into distinct regions. Moreover, our Gaussian Mixture Model (GMM) prior is adaptively updated with the partition result $\mathbf{H}$, which leverages Eq. (12) to further push towards learning a well-separated clustering structure (*i.e.*, a sharper distribution).

**Adversarial Regularizer**

The last part in our AGAE model is given by generative adversarial training [Goodfellow *et al.*, 2014], which directly regards our encoder network as the generator $G$, and employs a multi-layer neural network $f(\mathbf{z};\psi)$ as the discriminator $D$. Following [Makhzani *et al.*, 2016], we also conduct a min-max adversarial game between $D$ and $G$, where $D$ is trained to distinguish between the samples from $p(\mathbf{Z}|\mathbf{H})$ and $q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{S})$, while $G$ tries to fool the discriminator to thinking the encoded representations in Eq. (8) are sampled from the given prior in Eq. (11).

The adversarial regularizer of AGAE is defined with discriminative loss $\mathcal{L}_D$ and generative loss $\mathcal{L}_G$ as the following:

$$\mathcal{L}_D = \sum_{i=1}^{N} (\log f(\tilde{\mathbf{Z}}_i; \psi) + \log(1 - f(\mathbf{Z}_i; \psi)) \tag{13}$$

$$\mathcal{L}_G = \sum_{i=1}^{N} -\log f(\mathbf{Z}_i; \psi), \tag{14}$$

where $\tilde{\mathbf{Z}}$ represents the latent representations sampled from $p(\mathbf{Z}|\mathbf{H})$, and $\mathbf{Z}$ is given by $q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{S})$ through Eq. (6-8).

Upon the adversarial training, we except to leverage the discriminator $D$ to regularize the training of our inference model (*i.e.*, $q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{S})$ to learn a good graph-structured feature embedding space. This adversarial regularizer effectively avoids model over-fitting and leads to robust graph representations [Dai *et al.*, 2018; Pan *et al.*, 2018]. Moreover, it could better match the given prior compared with regularizing the encoded distribution with Kullback–Leibler divergence [Makhzani *et al.*, 2016]. The entire training procedure of our AGAE model is summarized by Algorithm 1.

# 3 Experiment

## 3.1 Experiment Setting

**Datasets.** Table 1 summarizes the statistics of all these datasets. We briefly introduce each dataset as follows. *1) Low-level features*: Pendigits[1] consists of 0-9 pen-written numbers, USPS[2] is a handwritten digit database of 16×16 images, Classic[3] collects abstracts from 4 sources related to information retrieval, and TDT2 [Cai *et al.*, 2009] is a document corpus of 30 topics. *2) Middle-level features*: Dslr [Kulis *et al.*, 2011] dataset collects office environment images taken by a DSLR camera, each of which is presented by a SURF BoW histogram feature. *3) High-level features:* AWA4K is a subset of Animal with Attributes (AWA) dataset [Lampert *et al.*, 2009], which includes 50 kinds of animal images. We randomly select 80 images from each class and eventually get 4000 images in total. FRGC is a three-dimension (3D) face image dataset, where we use the subset of FRGC with deep convolutional features provided by [Yang *et al.*, 2016]. STL-10[4] is a dataset of 96×96 RGB images from 10 objects, which provides 13K labeled images. In the experiment, we employ the VGG network [Simonyan and Zisserman, 2014] pre-trained on ImageNet as a feature extractor to encode each image in AWA4K and STL-10.

**Baseline Methods.** We compare the proposed AGAE model with three kinds of methods. 1) Two traditional clustering algorithms such as K-means and Spectral Clustering (SC) [Ng *et al.*, 2001]; 2) Four state-of-the-art ensemble clustering (EC) methods, including K-means based Consensus clustering (KCC) [Wu *et al.*, 2015], Spectral Ensemble Clustering (SEC) [Liu *et al.*, 2015], Infinite Ensemble Clustering (IEC) [Liu *et al.*, 2016], and Simultaneous Clustering and Ensemble (SCE) [Tao *et al.*, 2017a]. 3) Five recent deep embedding models, such as Adversarial Auto-Encoder (AAE) [Makhzani *et al.*, 2016], Variational Graph Auto-Encoder (VGAE) [Kipf and Welling, 2016], Deep Embedding Clustering [Xie *et al.*, 2016], Deep Clustering Network (DCN) [Yang *et al.*, 2017], and Adversarially Regularized Variational Graph Autoencoder (ARVGA) [Pan *et al.*, 2018]. All the compared methods are fed with the same features and tested by giving the true cluster number $K$. For all the EC methods, we generate Basic Partitions (BPs) with *random parameter selection* strategy [Fred and Jain, 2005]. Specifically, given $N$ data samples, we obtain $M$ BPs by running K-means $M$ times with randomly selecting cluster number from $[K, \sqrt{N}]$. We implement AAE according to [Makhzani *et al.*, 2016], with encoder/decoder using two hidden layers, and obtain the clustering result by running K-means with the output of the bottleneck layer. To make VGAE and ARGA adapt to generic data, we run the authors' code with recommended parameters and take as input a $k$-NN graph as following [Defferrard *et al.*, 2016], where the number of nearest neighbors is set to be 8 and the edges are weighted with Gaussian kernel. For DCN and DEC, we directly running the codes provided by the authors, and tune the parameters by following authors' suggestion. Two quantitative metrics are used to evaluate the clustering performance, which are *Average Clustering Accuracy (ACC)* and *Normalized Mutual Information (NMI)*. Both *ACC* and *NMI* are positive metrics ranged from 0 to 1, where a higher value indicates better performance.

| Dataset | #Instance | #Class | #Feature | Type |
|---|---|---|---|---|
| AWA4K | 4000 | 50 | 4096 | Image |
| Dslr | 157 | 10 | 800 | Image |
| FRGC | 2462 | 20 | 160 | Image |
| Pendigits | 10992 | 10 | 16 | Image |
| STL-10 | 13000 | 10 | 4096 | Image |
| USPS | 9298 | 10 | 256 | Image |
| Classic | 7094 | 4 | 41681 | Text |
| TDT2 | 9394 | 30 | 36771 | Text |

Table 1: Dataset details

**Implementation Details.** The proposed AGAE was implemented with TensorFlow toolbox. In Eq. (2), we set $\tau = 0.4$ as default. In our model, we employed a two-layer GCN network as the probabilistic encoder (*i.e.*, the generator), and a two-layer MLP network as the discriminator, where we set network dimensions of encoder as $d$-64-32 (*i.e.*, $m_1$=64 and $m_2$=32) and discriminator as 32-128-1. We set the activation function of all the hidden layers as Relu function, yet omitted the activation function of the top layer for encoder and used sigmoid function for the top layer of the discriminator network. The weights of all the layers were initialized by Xavier [Glorot and Bengio, 2010] approach. We adopted Adam [Kingma and Ba, 2014] with the default hyper-parameters as our optimizer, and conducted full-batch gradient descent to hold the structure of input graph. To achieve a good initialization and avoid random cluster assignment, we first pre-train our model with the fixed Gaussian prior

[1]https://archive.ics.uci.edu/ml/datasets

[2]http://www.cad.zju.edu.cn/home/dengcai/

[3]http://glaros.dtc.umn.edu/gkhome/

[4]https://cs.stanford.edu/~acoates/stl10/

| Methods | AGAE | Deep Embedding Methods | | | | | Ensemble Clustering Methods | | | | Baselines | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | DCN | DEC | VGAE | AAE | ARVGA | SCE | IEC | SEC | KCC | K-means | SC |
| AWA4K | **0.6790** | 0.3910 | 0.4092 | 0.5340 | *0.6553* | 0.6002 | 0.5952 | 0.5743 | 0.6467 | 0.6182 | 0.6244 | 0.6306 |
| Dslr | **0.5554** | 0.3950 | 0.4319 | 0.4420 | 0.3882 | 0.3911 | 0.5016 | 0.5089 | *0.5290* | 0.5229 | 0.4137 | 0.4293 |
| FRGC | **0.5114** | 0.4000 | 0.4124 | 0.4712 | 0.4976 | *0.5084* | 0.4665 | 0.4691 | 0.4529 | 0.4519 | 0.4638 | 0.4926 |
| Pendigits | **0.7817** | 0.7200 | 0.5640 | *0.7556* | 0.7314 | 0.6605 | 0.7092 | 0.7156 | 0.7109 | 0.6049 | 0.7108 | 0.7088 |
| STL-10 | **0.9325** | 0.8930 | 0.8413 | 0.8842 | 0.8653 | *0.9168* | 0.6694 | 0.6240 | 0.7976 | 0.7278 | 0.7723 | 0.8558 |
| USPS | *0.7357* | 0.6085 | **0.7762** | 0.6617 | 0.7121 | 0.6634 | 0.7319 | 0.6145 | 0.6815 | 0.5607 | 0.6320 | 0.6809 |
| Classic | **0.8674** | 0.6769 | *0.7990* | 0.5839 | 0.7512 | 0.6465 | 0.6713 | 0.5640 | 0.5066 | 0.4606 | 0.6644 | 0.6765 |
| TDT2 | *0.6052* | 0.4279 | 0.4178 | 0.4417 | 0.2346 | **0.6104** | 0.4644 | 0.4475 | 0.4510 | 0.4511 | 0.3357 | 0.3000 |

Table 2: Clustering performance on eight datasets by *ACC* (**red** color denotes the best and *blue* the runner-up)

| Methods | AGAE | Deep Embedding Methods | | | | | Ensemble Clustering Methods | | | | Baselines | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | DCN | DEC | VGAE | AAE | ARVGA | SCE | IEC | SEC | KCC | K-means | SC |
| AWA4K | **0.7669** | 0.4682 | 0.5698 | 0.6917 | 0.7232 | 0.7345 | 0.7393 | 0.7444 | *0.7601* | 0.7557 | 0.7490 | 0.7242 |
| Dslr | **0.5841** | 0.3680 | 0.4449 | 0.4403 | 0.3662 | 0.3810 | 0.5415 | 0.5456 | 0.5548 | *0.5628* | 0.4212 | 0.4117 |
| FRGC | **0.6493** | 0.4536 | 0.4933 | 0.5930 | 0.6309 | *0.6381* | 0.5916 | 0.6288 | 0.6160 | 0.6086 | 0.6150 | 0.6349 |
| Pendigits | **0.7415** | 0.6900 | 0.5938 | 0.7165 | 0.6856 | 0.6839 | 0.7017 | 0.7244 | *0.7270* | 0.6604 | 0.6870 | 0.6632 |
| STL-10 | **0.8741** | 0.8163 | 0.8578 | 0.8258 | 0.7796 | *0.8608* | 0.7924 | 0.7446 | 0.7830 | 0.7873 | 0.8153 | 0.8241 |
| USPS | *0.7415* | 0.5943 | **0.7996** | 0.6551 | 0.6018 | 0.7301 | 0.6863 | 0.6266 | 0.6529 | 0.6385 | 0.5954 | 0.6447 |
| Classic | **0.6477** | 0.4699 | *0.6226* | 0.5595 | 0.5749 | 0.3169 | 0.5028 | 0.4623 | 0.4006 | 0.3427 | 0.4336 | 0.4667 |
| TDT2 | *0.7632* | 0.5345 | 0.4780 | 0.6793 | 0.3668 | **0.7809** | 0.7280 | 0.6929 | 0.7083 | 0.6994 | 0.4017 | 0.5098 |

Table 3: Clustering performance on eight datasets by *NMI* (**red** color denotes the best and *blue* the runner-up)

and then finetune the network with our adaptive prior, as following the similar optimization strategy in [Xie *et al.*, 2016; Yang *et al.*, 2017; Li *et al.*, 2017]. We performed 200 epochs pre-training and fine-tuning on the USPS and STL-10 dataset, and 50 epochs for the remainder. The learning rate was set to be 1e-3 for pre-training and decayed to 1e-4 in fine-tuning.

### 3.2 Clustering Performance

Table 2 and Table 3 summarize the clustering performance of our approach and 11 strong baseline methods. As can be seen, we achieve the best performance in the majority cases, which shows the effectiveness of AGAE for clustering task.

*Compared with ensemble clustering*. As shown in Table 2 and Table 3, there are two important observations between ensemble clustering and other clustering algorithms. First, EC methods show impressive clustering performance on several datasets, such as on AWA4k and Dslr, even compared with recent deep clustering algorithms. This demonstrates the great potential of taking partition information as an effective supervision signal, and also justifies our motivation to learn a good graph embedding feature with consensus graph. Second, ensemble clustering methods degrade badly on some cases. For example, KCC, SEC and IEC all perform worse than traditional clustering methods on STL-10 and Classic. This is mainly due to the fact that ensemble clustering only takes as input multiple basic partitions, yet ignores the rich information from original features. To alleviate this problem, our approach jointly encodes features and consensus graph within a graph auto-encoder framework. The comparison results between AGAE and other EC methods demonstrate the superiority of using two-source information.

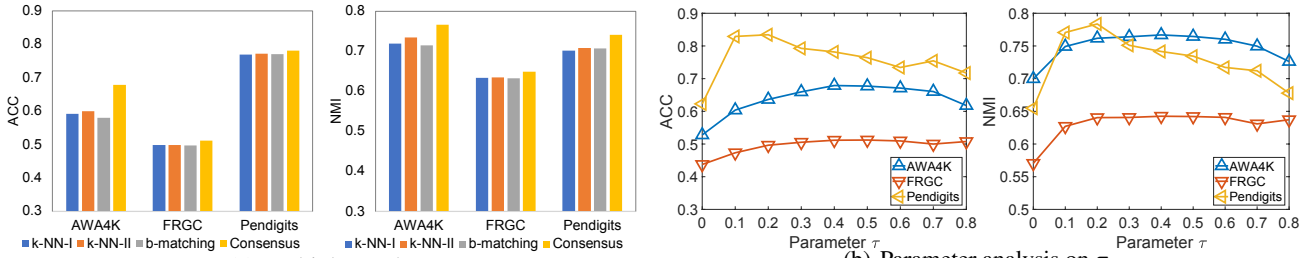
*Compared with deep embedding*. AAE [Makhzani *et al.*, 2016], VGAE [Kipf and Welling, 2016] and ARVGA [Pan *et al.*, 2018] are three recent deep generative models which mainly learn a probability distribution over embedding space via data/graph reconstruction; while DEC [Xie *et al.*, 2016] and DCN [Yang *et al.*, 2017] are two deep clustering models that simultaneously conduct feature learning and clustering process. Compared with deterministic mapping, generative model can better reveal data structure in the latent space, whilst enhance model generalization. As can be seen, AAE, VGAE and ARVGA achieve comparable results and sometimes even perform better than DEC and DCN, which indicates generative model could be a powerful alternative for the deep clustering task. On another hand, VGAE and ARVGA achieve competitive results compared to AAE with the utilization of graph structure. Different from these methods, our approach encapsulates ensemble clustering in a deep embedding process, by jointly using adversarial regularizer and GCN network, as well as an adaptive partition-dependent prior. As a result, the proposed AGAE achieves a clear improvement on the majority case.

### 3.3 Model Discussion

**Ablation Study.** The proposed AGAE introduces the adversarial training as a regularizer for the Graph Auto-Encoder (GAE) network, which is able to guide the probabilistic encoder to disentangle the latent space. We carry out an ablation study to validate the efficacy of our adversarial regularizer. Specifically, we compare our model with a vanilla GAE (*i.e.*, GCN based encoder and inner product decoder by only training with $\mathcal{L}_{AE}$ in Eq. (10)) and VGAE [Kipf and

| Methods | *Average Clustering Accuracy (ACC)* | | | | | *Normalized Mutual Information (NMI)* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AWA4K | Dslr | Pendigits | TDT2 | *avg* | AWA4K | Dslr | Pendigits | TDT2 | *avg* |
| GAE (GCN w/o regularizer) | 0.6410 | *0.5363* | 0.6761 | *0.5470* | 0.6001 | 0.7381 | 0.5566 | 0.7010 | *0.7451* | 0.6852 |
| VGAE (GCN + KL regularizer) | *0.6557* | 0.5325 | *0.7193* | 0.5431 | *0.6127* | *0.7491* | *0.5601* | *0.7170* | 0.7440 | *0.6926* |
| AGAE (GCN + adversarial regularizer) | **0.6790** | **0.5554** | **0.7817** | **0.6052** | **0.6553** | **0.7669** | **0.5841** | **0.7415** | **0.7632** | **0.7139** |
| AGAE over GAE | 0.0380 | 0.0191 | 0.1056 | 0.0582 | 0.0552 | 0.0288 | 0.0275 | 0.0405 | 0.0181 | 0.0287 |
| AGAE over VGAE | 0.0233 | 0.0229 | 0.0624 | 0.0621 | 0.0427 | 0.0178 | 0.0240 | 0.0245 | 0.0192 | 0.0214 |

Table 4: Ablation study on four datasets by *NMI* and *ACC* (**red** color denotes the best and *blue* the runner-up)




(a) Multiple graph (b) Parameter analysis on $\tau$

Figure 2: Exploration of different input graphs to the proposed AGAE. We test AGAE with (a) multiple graph construction methods and (b) different thresholding parameters on datasets of AWA4K, FRGC and Pendigits.

Welling, 2016] (*i.e.*, variational GAE regularized by the KL divergence between approximate posterior distribution and isotropic Gaussian prior) on four datasets. For a fair comparison, all these three models are fed with the same consensus graph, and training with the same GCN network architecture. As shown in Table 4, two important observations could be made. 1) VGAE generally performs better than GAE, implying that Gaussian prior is useful to guide the learning of latent space. Thus, it is reasonable for AGAE to pre-train the network with Gaussian prior to seek for a good initialization for the partition-dependent prior. 2) AGAE clearly improves the clustering performance over GAE and VGAE. This provides a strong empirical evidence to support the effectiveness of our adversarial regularizer for the clustering task.

**Multiple Graphs.** The proposed AGAE could be used as a general model that works with different graphs. Following [Li and Fu, 2015], we test our algorithm with other three common graphs, including 1) $k$-NN-I and 2) $k$-NN-II graph, which construct graph by using $k = 5$ and $k = 8$ nearest neighbors with Gaussian kernel; 3) $b$-matching graph [Jebara *et al.*, 2009], which builds a weighted and balance graph. Fig. 2(a) shows the performance of AGAE under these three graphs and the consensus graph provided by Eq. (2). As can be seen, consensus graph generally performs better than the other graphs (especially by *NMI*), which again demonstrates the superiority of partition information to original features. On another hand, we can also achieve comparable results with other kinds of graphs, which shows the potentiality of using AGAE for the general graph learning task, as well as multi-view learning [Ding *et al.*, 2019; Zhao *et al.*, 2017; Tao *et al.*, 2017b].

**Thresholding Parameter.** We employ a thresholding operator (parameterized by $\tau$) to alleviate noises and outliers for constructing the consensus graph in Eq. (2). Fig. 2(b) shows the performance of AGAE on four datasets by varying $\tau$ from 0 to 0.8. Note that, $\tau = 0$ means we directly use the consensus graph given by Eq. (1). As can be seen, the performance of AGAE generally goes up in the first steps, gets steady when $\tau \in [0.3, 0.5]$, and degrades when $\tau \geq 0.6$. This indicates the thresholding operation is useful to highlight the cluster structure of consensus graph. On the other side, Fig. 2(b) demonstrates the robust nature of ensemble clustering. Even by setting $\tau$ with a high value, consensus graph can still hold a good structure.

## 4 Conclusions

In this paper, a novel model named Adversarial Graph Auto-Encoders (AGAE) was proposed to incorporate ensemble clustering into a deep graph embedding process, which is trained by consensus graph reconstruction under the guidance of an adversarial regularizer. By leveraging graph convolutional network, AGAE jointly encodes the input features and consensus graph to capture the distribution of latent variables; and meanwhile, adversarial training with an adaptive partition-dependent prior is performed to further disentangle the embedding space. Experiments conducted on eight real-world datasets showed the effectiveness of our approach compared with several state-of-the-art deep embedding and ensemble clustering methods. Extensive model discussions were provided to showcase the proposed AGAE as a general framework applicable to different graphs, and validate the superiority of adversarial regularizer over two strong baselines.

## Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. This research is supported in part by the NSF IIS award 1651902 and U.S. Army Research Office Award W911NF-17-1-0367.

## References


[Bruna *et al.*, 2013] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013.

[Cai *et al.*, 2009] D. Cai, X. Wang, and X. He. Probabilistic dyadic data analysis with local and global consistency. In *ICML*, pages 105–112, 2009.

[Dai *et al.*, 2018] Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. Adversarial network embedding. In *AAAI*, 2018.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3837–3845, 2016.

[Ding *et al.*, 2019] Zhengming Ding, Handong Zhao, and Yun Fu. *Learning Representation for Multi-View Data Analysis - Models and Applications*. Advanced Information and Knowledge Processing. Springer, 2019.

[Domeniconi and Al-Razgan, 2009] Carlotta Domeniconi and Muna Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Trans. Knowl. Discov. Data*, 2(4):17:1–17:40, 2009.

[Fred and Jain, 2005] A. L. N. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680. 2014.

[Henaff *et al.*, 2015] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[Jebara *et al.*, 2009] Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and b-matching for semi-supervised learning. In *ICML*, pages 441–448, 2009.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Kulis *et al.*, 2011] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, pages 1785–1792, 2011.

[Lampert *et al.*, 2009] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *CVPR*, 2009.

[Lancichinetti and Fortunato, 2012] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *CoRR*, abs/1203.6093, 2012.

[Li and Fu, 2015] Sheng Li and Yun Fu. Learning balanced and unbalanced graphs via low-rank coding. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1274–1287, 2015.

[Li *et al.*, 2017] J. Li, T. Zhang, W. Luo, J. Yang, X. Yuan, and J. Zhang. Sparseness analysis in the pretraining of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 28(6):1425–1438, 2017.

[Liu *et al.*, 2015] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu. Spectral ensemble clustering. In *SIGKDD*, pages 715–724, 2015.

[Liu *et al.*, 2016] H. Liu, M. Shao, S. Li, and Y. Fu. Infinite ensemble for image clustering. In *SIGKDD*, pages 1745–1754, 2016.

[Luo *et al.*, 2011] D. Luo, C. H. Q. Ding, H. Huang, and F. Nie. Consensus spectral clustering in near-linear time. In *ICDE*, pages 1079–1090, 2011.

[Makhzani *et al.*, 2016] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *ICLR*, 2016.

[Ng *et al.*, 2001] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[Pan *et al.*, 2018] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *IJCAI*, pages 2609–2615, 2018.

[Simonyan and Zisserman, 2014] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Strehl and Ghosh, 2003] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2003.

[Tao *et al.*, 2017a] Z. Tao, H. Liu, and Y. Fu. Simultaneous clustering and ensemble. In *AAAI*, pages 1546–1552, 2017.

[Tao *et al.*, 2017b] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. From ensemble clustering to multi-view clustering. In *IJCAI*, pages 2843–2849, 2017.

[v. d. Maaten and Hinton, 2008] L. v. d. Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[Wang *et al.*, 2018] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *AAAI*, pages 2508–2515, 2018.

[Wu *et al.*, 2015] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen. K-means-based consensus clustering: A unified view. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):155–169, 2015.

[Xie *et al.*, 2016] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[Yang *et al.*, 2016] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, June 2016.

[Yang *et al.*, 2017] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870, 2017.

[Zhao *et al.*, 2017] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017.