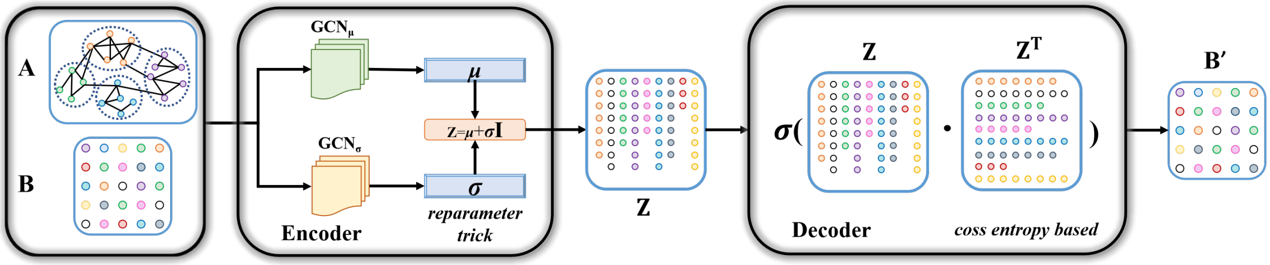

Figure 1: The architecture of the community detection model VGAER. We use the adjacency matrix A and modularity information B on the left (or concatenate the original features X of the nodes) to encode the mean vectors $\boldsymbol{\mu}$ and the standard deviation vectors $\boldsymbol{\sigma}$ of community memberships Z. The modularity matrix is reconstructed by the cross entropy based decoder on the right to maximize the modularity.

1 THEORETICAL ANALYSIS

1.1 Reconstruction and Modularity maximum

Modularity maximum model was first introduced by Newman to maximize the modularity index Q of the network [**?**], which is defined by the following:

$$Q=\frac{1}{2M}\sum_{i,j}\left[\left(a_{ij}-\frac{k_ik_j}{2M}\right)\mathcal{Z}\left(i,j\right)\right], \tag{1}$$

where M is the total number of edges of the network, a_{ij} is the adjacency matrix element, a value of 1 or 0 indicates whether there is a connected edge or not, k_i is the degree of node i, and $\mathcal{Z}$ is the association membership function of node i, when i and j belong to the same community, $\mathcal{Z}=1$, otherwise $\mathcal{Z}=0$.

Then, we simplify the Eq.(1) by defining the modularity matrix **B** and introducing the node community membership vectors. Define the modularity matrix $\mathbf{B}=[b_{ij}]$ as

$$b_{ij}=a_{ij}-\frac{k_ik_j}{2m}. \tag{2}$$

In this way, each node has a modularity relationship with all the other nodes, whether they have connected edges or not.

Next, we introduce a matrix $\mathbf{Z}=[z_{ij}]\in\mathbb{R}^{N\times K}$ which each row z_i is the community membership vector, and K is the dimension of the node community membership vector. So the Eq. (1) can be reduced as the following:

$$Q=\frac{1}{2m}\operatorname{Tr}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{B}\mathbf{Z}\right). \tag{3}$$

As a NP-hard problem, there are many different optimization ways to solve the maximization of Eq.(3). Here we introduce $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$ as the constant N condition to relax the problem, so we obtain the following modularity optimization problem after the relaxation:

$$\begin{gathered}\max Q=\max\left\{\operatorname{Tr}\left(\mathbf{Z}^{T}\mathbf{B}\mathbf{Z}\right)\right\}\\ \text{s.t. } \operatorname{Tr}\left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)=N.\end{gathered} \tag{4}$$

Based on the Rayleigh entropy, we know that the solution **Z** of the modularity degree maximization problem under relaxation conditions is the k largest eigenvectors of the modularity degree matrix **B**. Then, according to Eckart and Young's matrix reconstruction theorem [**?**], the equivalence of the modularity maximization and the modularity matrix reconstruction can be obtained.

LEMMA 1.1. *[Eckart and Young Theorem] For a matrix* D $\in\mathbb{R}^{m\times n}(m\geqslant n)$*, if* $\mathbf{D}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{T}$ *is the singular value decomposition of* $\mathbf{D}$*, and* $\mathbf{U},\mathbf{V},\Sigma=\operatorname{diag}\left(\sigma_1,\sigma_2,\ldots,\sigma_m\right)$ *where* $\sigma_1\geqslant\sigma_2\geqslant\ldots\geqslant\sigma_m$ *are as follows:*

$$\mathbf{U}=\left[\mathbf{U}_1\mathbf{U}_2\right],\Sigma=\left[\begin{array}{cc}\Sigma_1 & 0\\ 0 & \Sigma_2\end{array}\right],V=\left[\mathbf{V}_1\mathbf{V}_2\right] \tag{5}$$

where $\mathbf{\Sigma}_1$ *is* $r\times r,\mathbf{U}_1$ *is* $m\times r$ *and* $\mathbf{V}_1$ *is* $n\times r$*, then the optimal solution to the following problem*

$$\underset{\hat{\mathbf{D}}\in\mathbb{R}^{m\times n}\ \operatorname{rank}(\hat{\mathbf{D}})\leqslant r}{\operatorname{argmin}}\|\mathbf{D}-\hat{\mathbf{D}}\|_F \tag{6}$$

is $\hat{\mathbf{D}}^{*}=\mathbf{U}_1\Sigma_1\mathbf{V}_1^{T}$ *and*

$$\left\|\mathbf{D}-\hat{\mathbf{D}}^{*}\right\|_F=\sqrt{\sigma_{r+1}^2+\ldots+\sigma_m^2} \tag{7}$$

Theorem 1. *The modularity degree maximization problem under relaxation is equivalent to finding a k-order low-rank reconstruction of the modularity degree matrix* $\mathbf{B}$.

Proof. Note that $\mathbf{B}$ is a symmetric matrix, then $\mathbf{B}$ has an orthogonal decomposition $\mathbf{B} = \mathbf{X}\Lambda\mathbf{X}^T$. Under lemma 1.1, it can be further written as

$$\mathbf{B} = [\mathbf{H}, \mathbf{X}_1] \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} [\mathbf{H}, \mathbf{X}_1]^T \tag{8}$$

Λ_1 is the diagonal matrix corresponding to the former K-largest eigenvalue of B, and according to lemma **??**, $\mathbf{H}$ is the k-dimensional community membership vector of the network node, the optimal solution of the modular degree maximization problem under relaxation.

According to lemma 1.1, we set $\hat{\mathbf{B}} = \mathbf{H}\Lambda_1\mathbf{H}^T$, then $\hat{\mathbf{B}}$ is the best k-order low-rank reconstruction of $\mathbf{B}$. In summary, the modular degree maximization problem and the low-rank reconstruction problem of the module matrix are equivalent. □

Theorem 4 reveals the equivalence of modularity optimization problem and modularity matrix reconstruction, and further, we conduct a theoretical analysis of the reconstruction method, which is given by Theorems 2 and 3.

Theorem 2. *The K-order low-rank reconstruction matrix b of* $\mathbf{B}$ *satisfying the maximum modularity Q is a positive definite matrix.*

Proof. Firstly, because B is a symmetric matrix, there is an orthogonal decomposition; Secondly, according to Lemma 2, we can construct the best R-order low-rank approximation of B, and from Lemma 1, the first R feature vector is the optimal solution of the module degree optimization under the relaxation constraint, which is the community division of nodes.Here the proof that the reconstruction matrix Bhat is positive definite.

Without losing generality, we assume that Bhat is indefinite, that is, Bhat=H (with positive and negative eigenvalues) Ht, and Bhat is a low-rank approximation matrix that makes the module degree Q great. Bring into Lemma 1:

$$\boldsymbol{Q} = \operatorname{Tr}\left(\mathbf{H}^{\mathrm{T}}\mathbf{B}\mathbf{H}\right) \approx \operatorname{Tr}\left(\mathbf{H}^{\mathrm{T}}\hat{\mathbf{B}}\mathbf{H}\right) = \operatorname{Tr}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\Lambda\mathbf{H}^{\mathrm{T}}\mathbf{H}\right). \tag{9}$$

And H is orthogonal, so that $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}$, and thus

$$Q = \operatorname{Tr}(\Lambda) = \sigma_1 + \sigma_2 + \ldots + \sigma_r, \tag{10}$$

where $\sigma_1 > \sigma_2 > \ldots > \sigma_k > 0 > \ldots > \sigma_r$ □

Theorem 3. *The modularity maximal problem for community detection is equivalent to finding the best K-order low-rank reconstruction of the modular degree matrix B and can be written as* $\hat{\mathbf{B}} = \mathbf{Z}\mathbf{Z}^{\mathrm{T}}$.

Proof. The $\hat{\mathbf{B}}$ is known by Theorem 1 to be a positive-definite matrix, and $\hat{\mathbf{B}} = \mathbf{H}\Lambda\mathbf{H}^{\mathrm{T}}$, where $\mathbf{H}$ is the eigenvector corresponding to the eigenvalue with the largest K before $\mathbf{B}$, and Λ is a diagonal matrix composed of k eigenvalues. We can reconstruct B by the dot product by simply decomposing the triangles as $\Lambda = \Sigma^2$, where Σ is a diagonal matrix whose diagonal is the square of the triangular eigenvalues. So we have

$$\hat{\mathbf{B}} = \mathbf{H}\Sigma^2\mathbf{H}^{\mathrm{T}} = (\mathbf{H}\Sigma)\left(\Sigma\mathbf{H}^{\mathrm{T}}\right). \tag{11}$$

Let $\mathbf{Z}_B = \mathbf{H}\Sigma$, can get the dot product reconstruction form of modularity matrix B, and $\mathbf{Z}_B$ is just the low-rank encodings of the modularity $\mathbf{B}$ □

2 DIFFERENT ENCODING MODULES FOR VGAER

The essence of VGAER method is to obtain the low-order embedding probability of modularity (and feature connection) through the encoder based on graph neural network, and use this low-order embedding to reconstruct the distribution of modularity matrix. Therefore, VGAER allows us to use any GNN model or plug and play module in the encoding stage, such as: **GraphSAGE** [**?**], which realizes inductive learning through neighborhood sampling encoding and greatly reduces the complexity of the algorithm. We can see the extension of VGAER on GraphSAGE in the next section for the scalability; **GAT** [**?**], which enables VGAER to capture the weights between different node modules; **PAM** [**?**], which is actually a plug and play module with linear complexity, and faster incremental community detection can be achieved on the basis of GraphSAGE inductive learning; **GIN** [**?**], which introduce a ϵ to realize the injectivity of aggregation, thereby achieving a more powerful aggregation function.

Table 1: The NMI and Q values of networks on two different designed decoders

Dataset	VGAER		VGAER(dot)	
	NMI	Q	NMI	Q
Cora	0.447(↑ 59.1%)	0.657(↑ 3.2%)	0.281	0.636
Citeseer	0.227(↑ 187.3%)	0.610(↑ 7.8%)	0.079	0.566
Pubmed	0.273(↑ 565.9%)	0.558(↑ 66.1%)	0.041	0.336

3 DOES THE CROSS ENTROPY BASED DECODER REALLY WORK?

Then we demonstrate the effectiveness of the cross entropy based decoder through a set of simple experiments. As shown in the following Table 1, VGAER uses the cross entropy based decoder, and VGAER(dot) uses the 0-1 distributed dot product based decoder (following the design of VGAE).

The results in the Table 1 are interesting and non-trivial: firstly, the experiments on multiple datasets fully depict the effectiveness of the cross entropy based decoder we designed, and its NMI and Q on all datasets are significantly higher than those of the original VGAE's dot decoder. Secondly, the influence of the cross entropy based decoder on NMI (59.1%-565.9%) seems to be significantly greater than its influence on the Q values (3.2%-66.1%), which means that when the cross entropy based decoder is not used, most of the networks with ground truth will have a totally wrong division (NMI close to 0).

4 SCALABILITY

As a reconstruction method, the time complexity of the original VGAER is $O(MN)$ and the space complexity is $O(N^2)$. Therefore, we analyze the source of complexity in detail and introduce some techniques to enhance the scalability of VGAER.

We recommend the following techniques to enhance the scalability of VGAER. The time complexity of VGAER using the following methods will be reduced to $O(kN + pNF)$, where k is the number of neighborhood samples, p is the number of each batch, and F is the low dimensional attribution coding dimension.

Scalability techniques:

- k Neighbors sampling
- Mini-batch training
- Stochastic gradient descent

Now, we show the k neighbors sampling technique. We decompose one-step encoding into two-stage encoding: **Neighborhood Sharing** and **Membership Encoding**, through a limited number of neighbor sampling k; we will obtain condensed and high-quality neighborhood information.

The Neighborhood Sharing stage is as follow:

$$\mathcal{P}^l_{n(v)} = \mathcal{N}S\left(\left\{\mathcal{P}^{l-1}_u, \forall u \in \mathcal{N}(v)\right\}\right). \tag{12}$$

where $\mathcal{P} = \{\boldsymbol{\sigma}, \boldsymbol{\mu}\}$ is the set of variance and mean. $\mathcal{N}S$ is a Neighborhood Sharing operator. We use the MEAN as $\mathcal{N}S$, and $\mathcal{N}(v)$ represents a finite neighbors set of node v, and the number is usually k. The other operators are recommended as LSTM, Pool [**?**] and GAT [**?**] ect.

And the Membership Encoding stage is as follow:

When $l = 1$:

$$\mathcal{P}^1_v = \sigma\left(\text{CONCAT}\left(\mathcal{P}^0_v, \mathcal{P}^1_{\mathcal{N}(v)}\right) \cdot \mathbf{W}_0\right). \tag{13}$$

When $l = 2$:

$$\mathcal{P}^2_v = \text{CONCAT}\left(\mathcal{P}^1_v, \mathcal{P}^2_{\mathcal{N}(v)}\right) \cdot \mathbf{W}_i, \tag{14}$$

where $i = 1$ or 2 represents the second-layer weight matrix of variance and mean respectively.

The pseudo-code of the two-stage VGAER framework is showed in Algorithm 1.

Based on Algorithm 1, the more details of VGAER's mini-batch stochastic gradient training can be found in [**?**].

In short, with the use of these scalability technologies, the time complexity of each forward propagation of VGAER is $O(N)$, which is linear with the network nodes. However, if we want to further enhance the speed and scalability of VGAER (for example, the forward propagation time complexity reaches the sublinear or constant order $O(1)$), we must further consider the feasibility and method of compressing the input features.

5 SUPPLEMENTARY EXPERIMENT

We first report the Q or NMI values of each epoch on four networks to show the convergence of VGAER in Fig. **??**. It can be seen clearly from the Fig. **??** that the Q or NMI results of VGAER after convergence is high and stable, and the initial Q or NMI values of VGAER are relatively high.

Algorithm 1 Framework of the two-stage VGAER

```
Input: Graph $G(V,E)$; node features $\{\mathrm{x}_v, \forall v \in V\}$; layer depth $L$; weight matrices $\mathbf{W}_i, \forall i \in \{0,1,2\}$; non-linearity $\sigma$;
    neighborhood node set: $\mathcal{N}(v)$.
Output: Community membership vectors $z_v$ for all $\forall v \in V$
for $l$ =1,2 do
    for $\forall v \in V$ do
        Neighborhood Sharing: Sample the neighborhood codes of node $v$, and obtain neighbor information $\mathcal{P}^l_{\mathcal{N}(v)}$ by
        Eq.(12)
        Membership Encoding: Merge of $v$ and its neighborhood, and obtain low-rank membership encoding $\mathcal{P}^l_v$ by Eq.(13)
        and Eq.(14)
    end for
end for
$\mathbf{z}_v \leftarrow \boldsymbol{\mu}_v + \boldsymbol{\sigma}_v \circ \boldsymbol{\epsilon}_v, \forall v \in V$, $\{\boldsymbol{\mu}_v, \boldsymbol{\sigma}_v\} \in \mathcal{P}_v$
```

Then we report t-SNE [**?**] visualization on Power Grids and Cora. It can be seen from Fig. **??** that the embeddings of VGAE are mixed together and basically indistinguishable, which may be related to the reconstruction of the adjacency matrix **A**. While the embedding of VGAER forms a highly modular cluster structure, which is strongly distinguishable. The visualization results not only show the powerful modularity ability of VGAER, but also reveal the rationality of reconstructing the modularity matrix **B**. This can be understood from downstream tasks, because the original task of VGAE is link prediction, so the reconstruction of relationships between nodes is helpful. However the community detection task pays more attention to the higher-order relationship, and it may not be the most wise choice to continue to reconstruct the adjacency matrix (including VGAECD).

6 UNRELIABILITY OF SIMPLE RECONSTRUCTION FROM THE VARIATIONAL STATISTICS PERSPECTIVE

There is no doubt that a series of reconstruction methods represented by the autoencoder family have achieved unprecedented success in various fields of deep learning such as vision, speech, and text. We not only use reconstruction techniques to generate pseudo data, but also obtain the low-rank encodings from original data to deal with a wider range of downstream tasks such as node and link prediction, clustering, etc. But we have to point out that a number of studies have proved that low-rank codings are completely unidentifiable or random , that is, all parameters, including low-rank coding, are incapable of ensuring that the pseudo data of the autoencoder family is close to the original data. Guaranteed. We use Theorem 1 to formulate this conclusion and the proof of Theorem 1 is shown in the appendix. Before the Theorem 1, we give the symbol and definition at first. Consider a set of observation data (or graph signals) on the network $\mathbb{B} = \{\mathbf{b_1}, \mathbf{b_2}, ..., \mathbf{b_n}\}$, where b_i is the modularity information of the node (or the connection between modularity information and inherent features); consider the community membership encodes $\mathbb{Z} = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_n}\}$ obtained by VGAER or any deep generative model. Then we have the following

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\delta}}(\mathbf{b} \mid \mathbf{z}) p_{\boldsymbol{\delta}}(\mathbf{z}) = \int p_{\boldsymbol{\delta}}(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z}, \tag{15}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\boldsymbol{\theta}^*}(\mathbf{x}), \tag{16}$$

Theorem 4. *(unidentifiability) For all , for all observation and latent variable pairs (*$\mathbf{b}$*,*$\mathbf{z}$*), there is:*

$$\forall \left(\boldsymbol{\delta}, \boldsymbol{\delta}'\right): \quad p_{\boldsymbol{\delta}}(\mathbf{b}) = p_{\boldsymbol{\delta}'}(\mathbf{b}) \Longrightarrow \boldsymbol{\delta} = \boldsymbol{\delta}', \tag{17}$$

which is said that the autoencoder family are unidentifiable.

Theorem 5. *(unidentifiable) As a deep generative model, VGAER is always unidentifiable under unconditional situations.*

Proof. Without loss of generality,let . Now, a famous result shows that any orthogonal transformation of z has exactly the same distribution.

$$\begin{aligned} p_{\mathbf{z}'}(\boldsymbol{\xi}) &= p_{\mathbf{z}}\left(M^T \boldsymbol{\xi}\right) |\det M| = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\left\|M^T \boldsymbol{\xi}\right\|^2\right) \\ &= \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}\|\boldsymbol{\xi}\|^2\right) = p_{\mathbf{z}}(\boldsymbol{\xi}) \end{aligned} \tag{18}$$

where we have used the fact that the determinant of an orthogonal matrix is equal to unity. This result applies easily to any factorial prior. For z_i of any distribution, we can transform it to a uniform distribution by $F_i(z_i)$ where F_i is the cumulative distribution function of z_i. Next, we can transform it into standardized Gaussian by $\Phi^{-1}(F_i(z_i))$ where Φ is the standardized Gaussian cdf. After this transformation, we can again take any orthogonal transformation without changing the distribution. And we can even transform back to the same marginal distributions by $F_i^{-1}(\Phi(.))$. Thus, the original latents are not identifiable. □