Rethinking Variational Bayes in Community Detection From Graph Signal Perspective

Junwei Cheng, Yong Tang, Chaobo He, Pengxing Feng, Kunlin Han, and Quanlong Guan, *Member, IEEE*

***Abstract*—Methods based on variational bayes theorytare widely used to detect community structures in networks. In recent years, many related methods have emerged that provide valuable insights into variational bayes theory. Remarkably, a fundamental assumption remains incomprehensible. Variational bayes-based methods typically employ a posterior distribution that follows a gaussian distribution to approximate the unknown prior distribution. However, the complexity and irregularity of node distributions in real-world networks prompt us to consider what characteristics of network information are suitable for the posterior distribution. Mathematically, inappropriate low- and high-frequency signals in expectation inference and variance inference can intensify the adverse effects of community distortion and ambiguity. To analysis these two phenomena and propose reasonable countermeasures, we conduct an empirical study. It is found that appropriately compressing low-frequency signals during expectation inference and amplifying high-frequency signals during variance inference are effective strategies. Based on these two strategies, this paper proposes a novel variational bayes plug-in, namely VBPG, to boost the performance of existing variational bayes-based community detection methods. Specifically, we modulate the frequency signals during expectation and variance inference to generate a new gaussian distribution. This strategy improves the fitting accuracy between the posterior distribution and the unknown true distribution without altering the modules of existing methods. The comprehensive experimental results validate that methods using VBPG achieve competitive performance improvements in most cases.**

***Index Terms*—Variational bayes, community detection, graph signal process, graph neural networks, graph filters.**

Received 23 April 2024; revised 3 November 2024; accepted 14 February 2025. Date of publication 18 February 2025; date of current version 25 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62477016 and Grant 62077045, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011758, and in part by Scientific Research Innovation Project of Graduate School of South China Normal University under Grant 2024KYLX067. Recommended for acceptance by X. Zhu. *(Corresponding author: Yong Tang.)*

Junwei Cheng, Yong Tang, and Chaobo He are with the School of Computer Science, South China Normal University, Guangzhou 510000, China (e-mail: jung@m.scnu.edu.cn; ytang@m.scnu.edu.cn; hechaobo@m.scnu.edu.cn).

Pengxing Feng is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong 999077, China (e-mail: pengxfeng2-c@my.cityu.edu.hk).

Quanlong Guan is with the School of Information Science and Technology, Jinan University, Guangzhou 510000, China (e-mail: gql@jnu.edu.cn).

Kunlin Han is with the Computer Science Department, University of Southern California, Los Angeles, CA 90089 USA (e-mail: kunlinha@usc.edu).

The code is available at https://github.com/GDM-SCNU/VBPG.

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2025.3543378, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2025.3543378

I. Introduction

Community detection, as an important task in graph data mining, aims to divide nodes into optimal cluster structures with dense intra-edge and sparse inter-edge connections. In many fields, such as drug discovery [1] and educational data mining [2], researchers specifically exploit these cluster structures to advance their work. In recent years, emerging Graph Convolutional Networks (GCN), functioning as a graph filter, have demonstrated powerful data expression capabilities and become a mainstream technology for determining optimal cluster structures [3], [4], [5]. Motivated by GCN, variational bayes theory [6] is used in graph domain.

The Variational Graph Auto-Encoder (VGAE) [7], a classic method based on variational bayes theory, has attracted widespread attention. It consists of three main components: inference, generation and variational lower bound (ELBO), as shown in Fig. 1. Specifically, for a node x, its representation z is ideally determined by the probability density function of its true distribution. However, the unknown true distribution $p(z \mid x)$ prevents us from directly determining z. To address this problem, *a vanilla strategy is to introduce a posterior distribution* $q(z \mid x)$ *that follows a gaussian distribution.* Subsequently, the GCN samples z from the posterior distribution $q(z \mid x)$. Under the optimization of the scaled likelihood function (called ELBO), the posterior distribution gradually approximates the true distribution.

With the rise of variational bayes theory in community detection, many efforts have been directed towards improving the accuracy of detecting cluster structures. Qiu et al. [8] concatenates high-order modularity information with node attributes to *infer* the representation z. Grover et al. [9] attempts to *generate* the node information through an iterative graph refinement strategy. Meanwhile, some methods attempt to directly impose constraint conditions on representation z. For instance, Choong et al. [10] imposes a community assignment constraint by introducing the gaussian mixture model. Built on this method, GM-VGAE [11] imposes adversarial regularization on z. Obviously, the aforementioned works are centered around the benchmark framework (depicted in Fig. 1) and a posterior assumption. However, a fundamental assumption remains not well understood. The distribution of nodes in real-world networks is characterized by both complexity and irregularity. Despite relevant literature [12] shows that the gaussian distribution is suitable for fitting unknown distributions, a question arises: *What characteristics of network information does a gaussian distribution sample as*

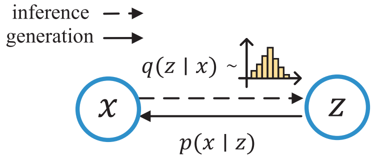


Fig. 1. A vanilla probabilistic framework for the variational bayes method in graph.

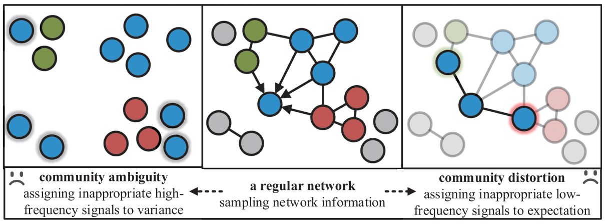

community ambiguity
assigning inappropriate high-frequency signals to variance
a regular network
sampling network information
community distortion
assigning inappropriate low-frequency signals to expectation

Fig. 2. A case of community anomalies, where the left represents community ambiguity, the middle represents a regular network, and the right represents community distortion. Nodes with the same color denote that they affiliate with the same community, and shadow indicates the original communities.

a posterior distribution to determine reasonable community assignments?

Mathematically, expectation of a gaussian distribution signifies the central tendency of the data distribution and reflects overall change. For nodes in the network, the central tendency of the data distribution is often reflected by low-frequency signals (i.e., similar features among nodes). To be specific, the use of a low-pass filter assigns cluster nodes with similar features to corresponding communities. Although the low-pass filter is the primary mechanism behind GCN [13], related literature [14] suggests that an excess of low-frequency signals can impact the anticipated representation z. One observation is that assigning inappropriate low-frequency signals to the expectation may result in the homogenization of expectations across communities. Specifically, the dissimilarity among communities may be weakened, increasing the likelihood that certain communities will erroneously annex multiple neighboring communities (see Fig. 2-right). We adapt the terminology of *community distortion* to refer to this problem.

In contrast, variance indicates the volatility of data and reflects individual differences. For each node in networks, individual differences are often reflected through high-frequency signals (i.e., node-specific features). Specifically, the high-pass filter helps to highlight the distinctiveness of each node by emphasizing distinct features while ignoring similarities. Although previous works [15], [16] have considered incorporating high-frequency signals into the learning process of representations, unthinkingly assigning these signals to variance may also result in negative effects. Generally, certain networks, such as social networks, have been confirmed to follow a power-law distribution and exhibit long-tail characteristics [17]. However, assigning accurate communities to tail nodes is challenging. This is due to their sparse connections and scarce attributes, which may lead them to exhibit consistent affiliation strengths with multiple communities. In this case, the inclusion of inappropriate high-frequency signals in the posterior distribution may result in excessive cumulative errors in the fitted distribution. This may cause the incorrect assignment of these nodes to a fixed community. An intuitive observation is that the community size is much larger than anticipated and lacks clear boundaries (Fig. 2-left). We adopt the terminology of *community ambiguity* to refer to this problem.

The existence of these two community anomalies prompts us to rethink more appropriate node characteristics for reconstructing a gaussian distribution. To explore the question, we start by conducting an empirical study (Section III). Our findings indicate that effectively mitigating the problems of community distortion and ambiguity involves appropriately compressing low-frequency signals in the expectation and amplifying high-frequency signals in the variance. To achieve this, a simple yet effective approach is to change frequency response functions [18]. However, this approach is considered naive because the large number of function expressions leads to a lack of generality. In recent years, some literatures [19], [20], [21] have shown that frequency components can be scaled by modifying certain network structures.

Inspired by empirical findings and previous works, this paper proposes a novel **V**ariational **B**ayes **P**lu**G**-in (VBPG), designed to boost the performance of existing community detection methods based on variational bayes theory. Specifically, we introduce two optimization objectives with variable limit function integration based on matrix perturbation theory. These objectives aim to scale the contributions of low- and high-frequency signals to the expectation and variance, respectively, thus mitigating the negative effects of community ambiguity and distortion on community detection results. Without altering the original modules of existing methods, the gaussian distribution reconstructed through VBPG fits the unknown true distribution more accurately. This enables more precise community assignments.

Previous works, such as [12], [22], [23], have primarily focused on optimizing posterior distributions without considering the distinct effects of low- and high-frequency signal reconstruction on community assignments. In contrast, we explicitly fill this gap through an empirical study. To the best of our knowledge, we are the first to analyze the impact of different signal reconstruction posterior distributions on community detection performance from a graph signal perspective in the context of variational Bayes. Additionally, based on the conclusions of our empirical study, we selectively modulate the frequency signals in variational inference. In contrast to existing methods [15], [16] that require the careful design of multiple filter functions to modulate node signals, our signal modulation strategy offers greater generalizability and applicability

The contributions of this paper are:

- *Analysis:* We conduct an empirical study to investigate the impact of posterior distribution from different signal reconstructions on community detection. By analyzing the experimental results, we summarize two rules to reduce the negative effects of community ambiguity and distortion on current methods.
- *Method:* We generalize two rules and further design a novel VBPG plug-in. Its function is to compress the

low-frequency signal during expectation inference and amplify the high-frequency signal during variance inference. Under the signal modulation of VBPG, the reconstructed gaussian distribution can more accurately fit the unknown true distribution.

- *Experiments:* We apply VBPG to twelve representative variational bayes-based community detection methods. Experimental results from five datasets demonstrate that integrating VBPG enables these methods to achieve competitive performance improvements without altering their modules.

The rest of this paper is organized as follows: Section II introduces the symbol definitions and related theories covered in this paper. Section III shows the comprehensive empirical study process and relted findings. Details on the proposed plug-in VBPG, along with the experimental results, are provided in Sections IV and V, respectively. Section VI briefly reviews related work. The paper concludes in Section VII.

II. Preliminaries

A. Community Detection

Given an undirected graph $G = (\mathbf{A}, \mathbf{H})$, where $\mathbf{A} \in \mathbb{R}^{N\times N}$ denotes the adjacency matrix with N nodes. For a pair of nodes u and v, if there is a relationship between them, $A_{u,v} = 1$, otherwise, $A_{u,v} = 0$. $\mathbf{H} \in \mathbb{R}^{N\times M}$ represents the attribute matrix with M attributes. The goal of community detection is to divide nodes into k disjoint communities $C = \{c_1, c_2, \ldots, c_k\}$ in an unsupervised manner.

B. Variational Bayes

Variational bayes-based methods aim to leverage the ELBO, obtained by Jensen's inequality, to match the posterior distribution with the true prior distribution. Depending on whether community constraints are directly imposed on the representation z, existing methods based on variational bayes can be broadly categorized into two groups: An end-to-end manner and a k-means based downstream manner. Given a node x, vanilla ELBO in the downstream manner is generalized as:

$$\log p(x) \geq \int_z q(z \mid y^x) \log p(x \mid z) dz - KL\left(q\left(z \mid y^x\right) \| p(z)\right) \tag{1}$$

where $y^x \rightarrow \{A_x, H_x, \ldots\}$ is the input set of x.

For the end-to-end manner, vanilla ELBO is generalized as:

$$\log p(x) \geq \int_z q(z \mid y^x) \left[\log \frac{p(x \mid z)p(z)}{q(z \mid y^x)} - KL\left(q(c \mid y^x) \| p(c \mid z)\right)\right] dz \tag{2}$$

where c signifies a non-specific community.

Despite slight variations in the ELBO between these two groups of methods, their inference and generation components can be summarized in a unified form. The inference component aims to determine the representation z via the posterior distribution $q(z \mid y^x)$. Without loss of generality, it is assumed to follow a gaussian distribution, whose expectation and variance are computed by two-layer GCN, as follows:

$$q(z \mid y^x) \sim \mathcal{N}\left(z \mid GCN_\mu(y^x), GCN_\sigma(y^x)\right) \tag{3}$$

The generation component generates the topological information of node x through a prior distribution $p(x \mid z)$. A common strategy is to consider the connection status of any set of node pairs to follow a bernoulli distribution. In this case, the prior distribution can be described as:

$$p(x \mid z) \sim Ber(z) \tag{4}$$

C. Graph Spectrum Theory

The graph spectrum theory is a core concept in graph signal processing. It relies on the eigenvalues and eigenvectors of the graph Laplacian matrix $\mathbf{L}$ to modulate the node signals. Let $\hat{\mathbf{L}}$ represents the symmetric normalized graph Laplacian matrix. It can be decomposed into the product of three matrices through eigendecomposition, i.e., $\hat{\mathbf{L}} = \mathbf{U}^T \Lambda \mathbf{U}$, where $\mathbf{U} = [u_1, u_2, \ldots, u_N]$ is the orthonormal eigenmatrix, and $\Lambda = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_N]$ is a diagonal matrix with eigenvalues $\lambda_i \in [0, 2]$ aligned in ascending order, where each λ_i corresponds to an eigenvector u_i. In graph signal processing, the low-frequency component, denoted by $\mathscr{F}^-$, corresponds to eigenvectors within the interval from λ_1 to $\lambda_{N/2}$ ($\lambda_{N/2} \approx 1$). Conversely, the high-frequency component $\mathscr{F}^+$ are linked to eigenvectors corresponding to eigenvalues within the interval from $\lambda_{(N/2)+1}$ to λ_N ($\lambda_N \approx 2$ [24]).

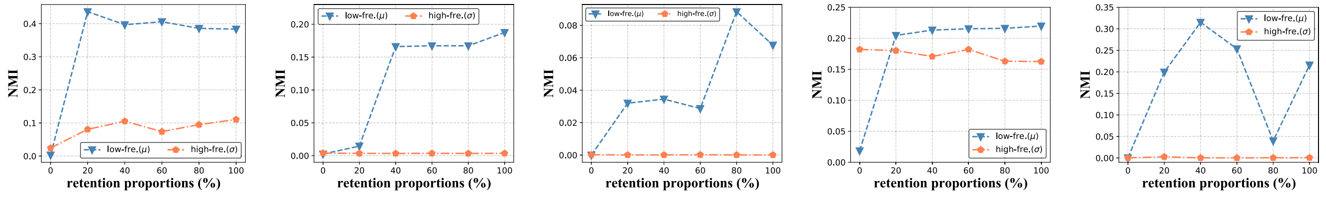
III. Empirical Study

In our empirical study, we investigate the impact of the posterior distribution of different signal reconstructions on community detection performance. Specifically, we modulate the contributions of low-frequency signals to the expectation and those of high-frequency signals to the variance. Building upon this foundation, we aim to summarize two rules that enable existing variational bayes-based methods to achieve reasonable performance improvements in community detection. Aiming at the phenomenon of community ambiguity and distortion, we will visually demonstrate them on a real network.

Considering the foundational role of VGAE in variational bayes methods, we employ it as an empirical method and conduct experiments on real-world datasets, including Cora, Citeseer, Pubmed, Blogcatalog and ACM. We utilize Normalized Mutual Information (NMI) as an evaluation metric to track the performance trends of VGAE, which employs different reconstructed gaussian distributions.

Given a symmetric normalized graph Laplacian matrix $\hat{\mathbf{L}}$, it can be decomposed into the product of eigenvalues and eigenvectors as follows:

$$\hat{\mathbf{L}} = \underbrace{(\lambda u u^T)_1 + \cdots + (\lambda u u^T)_{N/2}}_{\mathscr{F}^-} + \underbrace{(\lambda u u^T)_{1+(N/2)} + \cdots + (\lambda u u^T)_N}_{\mathscr{F}^+} \tag{5}$$



(a) Cora (b) Citeseer (c) Pubmed (d) Blogcatalog (e) ACM

Fig. 3. Performance of VGAE under different retention proportions of low-frequency/high-frequency signals.

For the expectation in a gaussian distribution, we retain low-frequency component in the proportions [0%, 20%, 40%, 60%, 80%, 100%], while the high-frequency component remains unchanged. Similarly, for the variance, we adopt the same approach and proportions to retain high-frequency component, leaving the low-frequency component unchanged. For example, if our goal is to retain 20% or discard 80% of the low-frequency component, then (5) can be adjusted to:

$$\hat{\mathbf{L}} \leftarrow \underbrace{(\lambda uu^T)_1 + \cdots + (\lambda uu^T)_{20\%\times(N/2)}}_{\mathscr{F}^-} + \mathscr{F}^+ \qquad (6)$$

Without loss of generality, we set all λ to 1, as the impact of different frequency response functions on the posterior distribution is not within the scope of our concern. The empirical results are presented in Fig. 3, and we get the following observations and insights:

Discarding some low-frequency signals during the expectation inference can improve the fit between the posterior distribution and the prior distribution: Observations from Fig. 3(a) indicate that discarding 80% of low-frequency signals enables VGAE to achieve optimal performance in experimental groups. As the discarded proportion decreases, performance correspondingly declines differently. Similar trends are observed in both Fig. 3(c) and (e). Among them, Fig. 3(e) is particularly prominent. Particularly, discarding 60% of low-frequency signals, compared to discarding 20%, results in a significant performance improvement, with a difference of nearly 28%. In the remaining two datasets, although the maximum values are obtained at the endpoints of the proportion interval, the suggestion to discard low-frequency signals remains valid. Specifically, discarding certain proportions of low-frequency signals, such as from 40% to 60% in Fig. 3(d) or (b), barely affects the performance of VGAE.

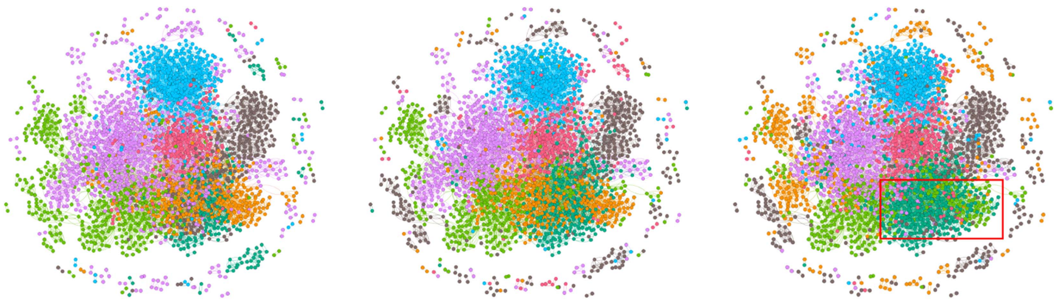
In theory, unlike ideal high-pass filters, appropriately discarding certain low-frequency signals typically does not affect core signal segments. These segments (usually $\lambda_i \to 0$) are crucial as they reflect the closeness between nodes and community centers. As illustrated in Fig. 3, discarding such signal segments, i.e., retention proportion is 0%, results in the collapse of all community structures within networks. With $\lambda_i \to 1$, homogeneous features of nodes become more prominent, ideally driving them closer to anticipated community centers. However, nodes characterized by topology sparsity and attribute scarcity may develop similar closeness to multiple community centers, resulting to community distortion.

To illustrate this phenomenon, we employ the Cora as an example to observe the impact of continuously discarding low-frequency signals during expectation inference on community structures. As depicted in Fig. 4(b), despite discarding 80% of the low-frequency signals, the community structures still closely approximate the true results. However, as the discarded proportion decreases, the accumulation of node homogeneity information results in partial community distortion in the network. In particular, the red box in Fig. 4(c) highlights a pronounced community distortion, in which the original community structure is almost completely engulfed.

Retaining more high-frequency signals during variance inference promotes fit accuracy between the posterior distribution and the prior distribution: Upon revisiting Fig. 3, it is observed that as the proportion of high-frequency signals increases, the performance of VGAE only changes slightly during variance inference. Specifically, across the six experimental groups within Cora and Blogcatalog, the data variance is recorded as 2.82% and 0.85%, respectively. For the remaining four datasets, it is difficult to track trends in performance changes.

Theoretically, the variance in a gaussian distribution reflects the distinctive characteristics of data itself. Although some nodes have lower proximity to the community centers, high-frequency signals help them form clear boundaries for some communities. However, the limited availability of high-frequency signals may cause these nodes to be incorrectly assigned to a fixed community, which we call community ambiguity. To illustrate this phenomenon, we employ the Cora as a case study, as detailed in Fig. 5. The results show that when the retention of high-frequency signals is low (e.g., 0%), the ratio of node membership in certain communities, such as c_1 and c_7, deviates significantly from their actual values. Conversely, as the retention proportion increases, the ratio of node membership in most communities gradually becomes consistent with the actual value.

We find that increasing the proportion of high-frequency signals contributes to alleviating the negative impact of community ambiguity on VGAE. However, the non-significant trends in Fig. 3 prompt us to consider the role of low-frequency signals in variance inference to support our conclusions. Fig. 6 shows the trend of the performance impact of altering low-frequency signals on VGAE during the variance inference. It is observed that the performance of VGAE generally decreases as the retention proportions increases, such as in Cora, Citeseer, ACM and Pubmed, or even approaching 0 in some cases. Although the performance trend observed on Blogcatalog differs, the



(a) true communities (b) retention 20% (c) retention 80%

Fig. 4. Community distortion under different retention proportions of low-frequency component. Note that nodes in the same community are painted with the same color.

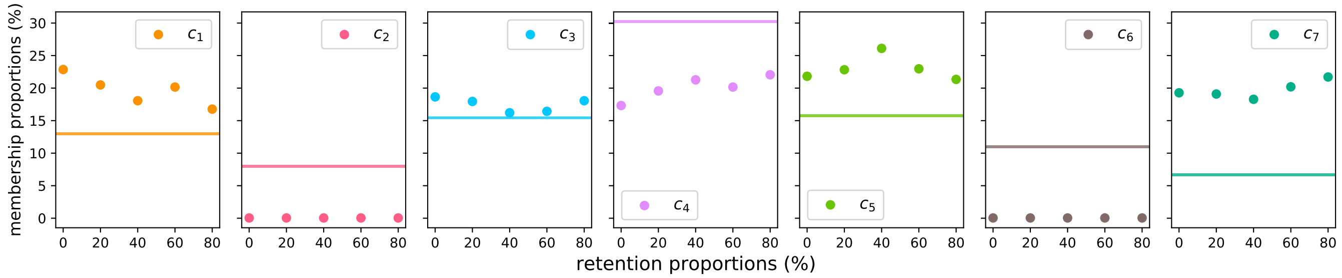


Fig. 5. Community ambiguity under different retention proportions of high-frequency component. Note that the lines represent the true proportions, while the scatter points represent the VGAE-calculated proportions.

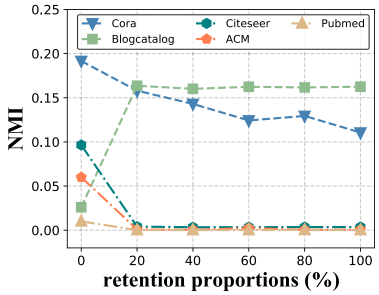


Fig. 6. Performance of VGAE under different retention proportions of low-frequency signals during variance inference.

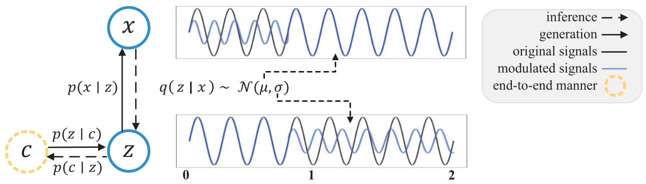


Fig. 7. Probabilistic framework based on variational bayes methods using the VBPG plug-in.

performance of VGAE exhibits only slight changes with increasing proportions. That is, retaining only a small portion of low-frequency signals already yields competitive results.

IV. Methodology

A. Overview

Before introducing the VBPG plug-in as shown in Fig. 7, we first outline our design ideas. The empirical study suggests that to minimize the negative effects of community distortion and ambiguity on method effectiveness, a straightforward approach is to discard some low-frequency signals in expectation inference and retain more high-frequency signals in variance inference. *However, simply discarding frequency signals is undesirable, as it would lead to the loss of potential network information and greatly diminish the performance of methods.*

To this end, we adopt signal modulation instead of discarding frequency signals directly, thus avoiding the loss of potential network information. Specifically, we introduce variable limit function integration to adjust the dividing points in the frequency components:

- In expectation inference, we employ a signal compression strategy to compress the low-frequency component, which helps to alleviate community distortion. As the iterative process advances, the upper limit of the integral for the low-frequency component will exhibit a minor adjustment towards $\lambda \rightarrow 0$. Simultaneously, the lower limit of the integral for the high-frequency component moves slightly in the same direction.
- In variance inference, we utilize a signal amplification strategy to enhance the presence of the high-frequency

component, which helps to mitigate the negative effect of community ambiguity. As the iterative procedure progresses, the lower limit of the integral for the high-frequency component gradually moves towards $\lambda \to 0$. Also, the upper limit of the low-frequency component moves in the same direction.

Mathematically, the compression strategy involved in expectation inference significantly enhances the closeness of nodes to community centers, particularly for nodes with homogeneous characteristics. The mechanism is to compress the redundant information within the original low-frequency segment by modulating signals. For the amplification strategy involved in variance inference, it helps to fine-tune the proximity of community centers to nodes characterized by topology sparsity and attribute scarcity. The mechanism involves expanding the frequency range within the high-frequency component through signal modulation. Thereby, the diversity features of nodes in the initial spectrum are amplified, which simultaneously enhances the ability of nodes to distinguish similar communities.

In summary, the proposed VBPG plug-in allows the reconstructed gaussian distribution to more accurately approximate the true unknown distribution by adjusting the dividing points within the frequency components. For most methods that adopt the gaussian distribution as their posterior assumption, the VBPG is capable to enhance the performance of these methods without altering the original modules, regardless of whether they are designed to detect communities in an end-to-end manner.

B. Expectation Inference

Based on empirical study findings and the generalization of conclusions, we compress the low-frequency component in the expectation inference by modifying the network structure. As described in (5), the matrix $\hat{\mathbf{L}}$ can be decomposed into low- and high-frequency components. Assuming the frequency response function is fixed, the compression of frequency signals can be interpreted as a perturbation of the eigenvalues. Taking the frequency response function $g(\lambda) = 1 - \lambda$ of GCN as an example, the eigenvalue $\lambda = 1$ is the critical point to distinguish low-frequency component from high-frequency component. For frequency segment with eigenvalues λ less than 1, they correspond to smoother frequency signals, which emphasize the similarity of features among nodes. Conversely, the frequency segment with eigenvalues λ greater than 1 indicates more significant differences in frequency signals, which highlights the distinct features among nodes. Therefore, in expectation inference, compressing the frequency signals is equivalent to mapping the eigenvalues around the critical point to a narrower range. This process also requires the expansion of the high-frequency component to maintain the continuity of frequency signals. In this study, we employ matrix perturbation theory to analyze the perturbation of eigenvalues and to formulate an optimization objective for expectation inference. Based on spectral graph theory, graph Laplacian matrix $\mathbf{L}$ is an $N-$order matrix, which satisfies the following conditions:

$$\mathbf{L}u = \lambda u \iff u^T\mathbf{L} = \lambda u^T \tag{7}$$

where u represents a eigenvector. Applying a perturbation to the Laplacian matrix, (7) is rewritten as:

$$\begin{aligned}&(\Delta\mathbf{L} + \mathbf{L})(u + \Delta u) = (\lambda + \Delta\lambda)(u + \Delta u)\\ \Rightarrow\ &\Delta\mathbf{L}u + \Delta\mathbf{L}\Delta u + \mathbf{L}u + \mathbf{L}\Delta u = \lambda u + \lambda\Delta u + \Delta\lambda u! + \ \Delta\lambda\Delta u\end{aligned} \tag{8}$$

$$\because \mathbf{L}u = \lambda u \tag{9}$$

$$\therefore \Delta\mathbf{L}u + \mathbf{L}\Delta u = \lambda\Delta u + \Delta\lambda u + \mathcal{O}(\Delta) \tag{10}$$

where $\mathcal{O}(\Delta)$ is a infinitesimal of higher order and Δ represents the subtle perturbation of matrix. We left-multiply each term in (10) by u^T to get:

$$u^T\Delta\mathbf{L}u + u^T\mathbf{L}\Delta u = u^T\lambda\Delta u + \Delta\lambda u^T u + \mathcal{O}(\Delta) \tag{11}$$

$$\because \mathbf{L}\Delta u = \lambda\Delta u \tag{12}$$

$$\therefore u^T\Delta\mathbf{L}u + \lambda u^T\Delta u = \lambda u^T\Delta u + \Delta\lambda u^T u + \mathcal{O}(\Delta) \tag{13}$$

$$\begin{aligned}&\therefore u^T\Delta\mathbf{L}u = \Delta\lambda + \mathcal{O}(\Delta)\\ &\Rightarrow \Delta\lambda \approx u^T\Delta\mathbf{L}u\end{aligned} \tag{14}$$

Consequently, the specific equation of eigenvalue perturbation under matrix perturbation theory is as follows:

$$\Delta\lambda = \lambda - \lambda^- = \lambda^+ - \lambda \approx u^T\Delta\mathbf{L}u \tag{15}$$

where λ^- and λ^+ denote the modified eigenvalues toward the low- and high-frequency components, respectively.

For a given eigenvalue λ_i in low-frequency component, its change under matrix perturbation is expressed as follows:

$$|\Delta\lambda_i| = |\left(u^T\Delta\mathbf{L}u\right)_i| = (u^T\Delta\mathbf{A}_\mu^- u)_i - \left(u^T\Delta\mathbf{D}_\mu^- u\right)_i \tag{16}$$

where $\Delta\mathbf{A}_\mu^-$ represents the perturbation of the adjacency matrix in the low-frequency component, and $\Delta\mathbf{D}_\mu^-$ denotes the perturbation of the degree matrix corresponding to $\Delta\mathbf{A}_\mu^-$.

Without loss of generality, we assume the eigenvalues within the low-frequency component to be continuous variables. The compression of low-frequency signals is converted into an integral over the interval [0, θ], as detailed below:

$$\begin{aligned}\int_0^\theta g(\lambda)|\Delta\lambda|d\lambda &= \int_0^\theta g(\lambda)\left[\left(u^T\Delta\mathbf{A}_\mu^- u\right) - \left(u^T\Delta\mathbf{D}_\mu^- u\right)\right]d\lambda\\ &\gtrapprox \int_0^\theta g(\lambda)\left(u^T\Delta\mathbf{A}_\mu^- u\right)d\lambda - \int_0^\theta g(\lambda)N d\lambda\\ &\gtrapprox \int_0^\theta g(\lambda)\left(u^T\Delta\mathbf{A}_\mu^- u\right)d\lambda - \theta N\end{aligned} \tag{17}$$

where $\theta \leq 1$. $g(\lambda)$ is defined as the frequency response function that, given a specific eigenvalue, generates the corresponding amplitude.

Simultaneously, to ensure the continuity of frequency signals, the amplification of high-frequency component is converted into integration over the calculation interval [θ,2], as follows:

$$\begin{aligned}\int_\theta^2 g(\lambda)|\Delta\lambda|d\lambda &= \int_\theta^2 g(\lambda)\left[\left(u^T\Delta\mathbf{D}_\mu^+ u\right) - \left(u^T\Delta\mathbf{A}_\mu^+ u\right)\right]d\lambda\\ &\lessapprox \int_\theta^2 \theta N - g(\lambda)(u^T\Delta\mathbf{A}_\mu^+ u)d\lambda\end{aligned} \tag{18}$$

Combining the above integral interval, the optimization objective for expectation inference is reformulated as minimizing the integral over the interval [0, 2], as follows:

$$\begin{aligned}
\min \mathcal{L}_\mu &= \min \int_0^2 g(\lambda)|\Delta\lambda| d\lambda \\
&= \min \int_0^\theta g(\lambda)|\Delta\lambda| d\lambda - \max \int_\theta^2 g(\lambda)|\Delta\lambda| d\lambda \\
&\approx \int_0^\theta g(\lambda) \left(u^T \Delta\mathbf{A}_\mu^- u\right) d\lambda - \int_\theta^2 g(\lambda) \left(u^T \Delta\mathbf{A}_\mu^+ u\right) d\lambda \\
&= \int_0^\theta \sum u^T g(\lambda) u \times \Delta\mathbf{A}_\mu^- d\lambda \\
&\quad - \int_\theta^2 \sum u^T g(\lambda) u \times \Delta\mathbf{A}_\mu^+ d\lambda
\end{aligned} \tag{19}$$

note that u and $g(\lambda)$ are different in the two integrals and their superscripts have been omitted to clarify the equation.

Here, we need to impose guiding constraints on the optimization objective of expectation inference. On the one hand, we aim to avoid losing the characteristics of the graph structure when processing frequency signals. On the other hand, we hope the frequency signals can reflect the community membership information. Inspired by previous works [21], [25], [26], the final optimization objective of expectation inference is as follows:

$$\begin{aligned}
&\min \mathcal{L}_\mu \\
&\approx \int_0^\theta \sum u^T g(\lambda) u \times \Delta\mathbf{A}_\mu^- d\lambda \\
&\quad - \int_\theta^2 \sum u^T g(\lambda) u \times \Delta\mathbf{A}_\mu^+ d\lambda \\
&\quad + \sigma H(\Delta\mathbf{A}_\mu^-) + \sum f \times (\Delta\mathbf{A}_\mu^- - \mathcal{C}) + \sum s \\
&\quad \times \left[\left(\Delta\mathbf{A}_\mu^-\right)^T - \mathcal{C}\right] \\
&\quad + \sigma H(\Delta\mathbf{A}_\mu^+) + \sum f \times (\Delta\mathbf{A}_\mu^+ - \mathcal{C}) \\
&\quad + \sum s \times \left[(\Delta\mathbf{A}_\mu^+)^T - \mathcal{C}\right]
\end{aligned} \tag{20}$$

where $H(x) = -\sum x(\log x - 1)$, and σ is a hyperparameter that controls the smoothness balance of the frequency signal. $f \in \mathbb{R}^{N\times 1}$ and $s \in \mathbb{R}^{N\times 1}$ are lagrange multipliers. $\mathcal{C}$ is the intra-cluster distance, as follows:

$$\mathcal{C} = \frac{1}{k} \sum_{c \in C} \frac{1}{|c|(|c|-1)} \sum_{z_u z_v \in c, z_u \neq z_v} \|z_u - z_v\|_2 \tag{21}$$

C. Variance Inference

In variance inference, we adopt a same strategy to amplify high-frequency signals. Meanwhile, we compress the low-frequency component to maintain the continuity of the frequency signals. Specifically, according to the continuous variable assumption, the amplification of high-frequency signals is converted into an integral over the interval [θ, 2], as follows:

$$\begin{aligned}
\int_\theta^2 g(\lambda)|\Delta\lambda| d\lambda &= \int_\theta^2 g(\lambda) \left[\left(u^T \Delta\mathbf{D}_\sigma^+ u\right) - \left(u^T \Delta\mathbf{A}_\sigma^+ u\right)\right] d\lambda \\
&\lessapprox \theta N - \int_\theta^2 g(\lambda) \left(u^T \Delta\mathbf{A}_\sigma^+ u\right) d\lambda
\end{aligned} \tag{22}$$

where $\theta \leq 1$.

Additionally, to ensure the continuity of frequency signals, the compression of low-frequency component is converted into integration over the calculation interval [0, θ], as follows:

$$\begin{aligned}
\int_0^\theta g(\lambda)|\Delta\lambda| d\lambda &= \int_0^\theta g(\lambda) \left[\left(u^T \Delta\mathbf{A}_\sigma^- u\right) - \left(u^T \Delta\mathbf{D}_\sigma^- u\right)\right] d\lambda \\
&\gtrapprox \int_0^\theta g(\lambda) \left(u^T \Delta\mathbf{A}_\sigma^- u\right) d\lambda - \theta N
\end{aligned} \tag{23}$$

Combining the above integral interval, the optimization objective for variance inference is reformulated as maximizing the integral over the interval [0, 2], as follows:

$$\begin{aligned}
\max \mathcal{L}_\sigma &= \max \int_0^2 g(\lambda)|\Delta\lambda| d\lambda \\
&= \max \int_\theta^2 -g(\lambda)|\Delta\lambda| d\lambda - \min \int_0^\theta g(\lambda)|\Delta\lambda| d\lambda \\
&\approx \int_\theta^2 -g(\lambda)(u^T \Delta\mathbf{A}_\sigma^+ u) d\lambda + \int_0^\theta -g(\lambda)(u^T \Delta\mathbf{A}_\sigma^- u) d\lambda \\
&= \int_\theta^2 -\sum u^T g(\lambda) u \times \Delta\mathbf{A}_\sigma^+ d\lambda \\
&\quad + \int_0^\theta -\sum u^T g(\lambda) u \times \Delta\mathbf{A}_\sigma^- d\lambda
\end{aligned} \tag{24}$$

Here, we impose the similar constraints as expectation inference, the final optimization objective is as follows:

$$\begin{aligned}
&\max \mathcal{L}_\sigma \approx \int_\theta^2 \\
&-\sum u^T g(\lambda) u \times \Delta\mathbf{A}_\sigma^+ d\lambda + \int_0^\theta -\sum u^T g(\lambda) u \times \Delta\mathbf{A}_\sigma^- d\lambda \\
&+ \beta H\left(\Delta\mathbf{A}_\sigma^+\right) + \sum f \times \left(\Delta\mathbf{A}_\sigma^+ - \mathcal{C}\right) \\
&+ \sum s \times \left[\left(\Delta\mathbf{A}_\sigma^+\right)^T - \mathcal{C}\right] \\
&+ \beta H\left(\Delta\mathbf{A}_\sigma^-\right) + \sum f \times \left(\Delta\mathbf{A}_\sigma^- - \mathcal{C}\right) \\
&+ \sum s \times \left[\left(\Delta\mathbf{A}_\sigma^-\right)^T - \mathcal{C}\right]
\end{aligned} \tag{25}$$

where β is a hyperparameter that controls the smoothness balance of the frequency signal.

D. Optimization of VBPG

We combine the geometric meaning of integrals with the approach of solving extreme values through derivatives, providing approximate solutions to objective (20) and (25). Here, we use the compression of low-frequency signals in the expectation inference as an example to demonstrate our solution. For simplicity, we only keep the expressions in (20) that involve compression of low-frequency signals and ignore some subscripts, as follows:

$$\begin{aligned}\mathcal{L}_{\mu} \approx & \int_0^{\theta} \sum u^T g(\lambda) u \times \Delta\mathbf{A}^- d\lambda + \alpha H\left(\Delta\mathbf{A}^-\right) \\ & + \sum f \times \left(\Delta\mathbf{A}^- - \mathcal{C}\right) + \sum s \times \left[(\Delta\mathbf{A}^-)^T - \mathcal{C}\right]\end{aligned} \tag{26}$$

Based on the geometric interpretation of the integral, the maximum value of the integral component in (26) cannot exceed the area of the rectangle, which is determined by $[\theta, 0]$ as the base and the integrand as the height. Meanwhile, the maximum value of the area cannot exceed the area of the square with $\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-$ as the length of each side. Consequently, it can be further scaled to:

$$\begin{aligned}\mathcal{L}_{\mu} \approx & (\theta - 0) \times \sum u^T g(\lambda) u \times \Delta\mathbf{A}^- + \alpha H\left(\Delta\mathbf{A}^-\right) \\ & + \sum f \times \left(\Delta\mathbf{A}^- - \mathcal{C}\right) + \sum s \times \left[\left(\Delta\mathbf{A}^-\right)^T - \mathcal{C}\right] \\ & \leq \left(\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-\right)^2 + \alpha H\left(\Delta\mathbf{A}^-\right) \\ & + \sum f \times \left(\Delta\mathbf{A}^- - \mathcal{C}\right) + \sum s \times \left[\left(\Delta\mathbf{A}^-\right)^T - \mathcal{C}\right]\end{aligned} \tag{27}$$

To solve for $\Delta\mathbf{A}^-$, we compute the partial derivative of $\Delta\mathbf{A}^-$ in $\mathcal{L}_{\mu}$ as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mu}}{\partial \Delta\mathbf{A}^-} \approx & 2 \times \left(\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-\right) \times \sum u^T g(\lambda) u \\ & - \alpha \log \Delta\mathbf{A}^- + f + s\end{aligned} \tag{28}$$

By setting (28) $= 0$, the extreme value of $\Delta\mathbf{A}^-$ can be expressed as:

$$\begin{aligned} & 2 \times \left(\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-\right) \times \sum u^T g(\lambda) u \\ & - \alpha \log \Delta\mathbf{A}^- + f + s = 0 \\ \Rightarrow & \Delta\mathbf{A}^- = e^{2/\sigma \times (\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-) \times \sum u^T g(\lambda) u} + e^{f/\sigma} + e^{s/\sigma} \\ \Rightarrow & \Delta\mathbf{A}^- = \mathbf{F}\mathbf{K}^-\mathbf{S}\end{aligned} \tag{29}$$

where $\mathbf{K}^- = e^{2/\sigma \times (\sum u^T g(\lambda) u \times \Delta\mathbf{A}^-) \times \sum u^T g(\lambda) u}$. $\mathbf{F} = diag(e^{f/\sigma})$ and $\mathbf{S} = diag(e^{s/\sigma})$, with Sinkhorn Iteration applied to solve for f and s.

Similarly, $\Delta\mathbf{A}^+$ is as follows

$$\Delta\mathbf{A}^+ = \mathbf{F}\mathbf{K}^+\mathbf{S} \tag{30}$$

where $\mathbf{K}^+ = e^{2/\sigma \times (\sum u^T g(\lambda) u \times \Delta\mathbf{A}^+) \times \sum u^T g(\lambda) u}$. Note that $\mathbf{F}$ and $\mathbf{S}$ are different in (29) and (30).

Finally, the modulation of signals in expectation inference is summarized as the adjustment to the adjacency matrix, as follows:

$$\mathbf{A}_{\mu} = \theta \times \Delta\mathbf{A}^- - (2 - \theta) \times \Delta\mathbf{A}^+ + \mathbf{A} \tag{31}$$

Note that in our VBPG plug-in, we set θ to gradually attenuate to ensure signal smoothness, starting from 1 and decreasing by 0.05 after every $\mathcal{T}$ epochs. Without loss of generality, we set $\mathbf{L} \leftarrow \sum u^T g(\lambda) u$ for all scenes.

In variance inference, we adopt the same approach to derive $\Delta\mathbf{A}_{\sigma}^+$ and $\Delta\mathbf{A}_{\sigma}^-$, as follows:

$$\begin{aligned} & 2 \times \left(-\sum u^T g(\lambda) u \times \Delta\mathbf{A}_{\sigma}^+\right) \times \sum u^T g(\lambda) u \\ & - \beta \log \Delta\mathbf{A}_{\sigma}^+ + f + s = 0 \\ \Rightarrow & \Delta\mathbf{A}_{\sigma}^+ = \mathbf{F}\mathbf{K}^+\mathbf{S}\end{aligned} \tag{32}$$

and

$$\begin{aligned} & 2 \times \left[-\sum u^T g(\lambda) u \times \Delta\mathbf{A}_{\sigma}^-\right] \times \sum u^T g(\lambda) u \\ & - \beta \log \Delta\mathbf{A}_{\sigma}^- + f + s = 0 \\ \Rightarrow & \Delta\mathbf{A}_{\sigma}^- = \mathbf{F}\mathbf{K}^-\mathbf{S}\end{aligned} \tag{33}$$

Finally, the modulation of signals in variance inference is summarized as the adjustment to the adjacency matrix, as follows:

$$\mathbf{A}_{\sigma} = (2 - \theta) \times \Delta\mathbf{A}_{\sigma}^+ - \theta \times \Delta\mathbf{A}_{\sigma}^- + \mathbf{A} \tag{34}$$

In summary, the node representation z will be determined by the reconstructed gaussian distribution, as follows:

$$z \sim \mathcal{N}(z \mid \mu, \sigma)) \tag{35}$$

where

$$\mu = GCN_{\mu}\left(GCN(y^x), \mathbf{A}_{\mu}, \ldots\right) \tag{36}$$

$$\sigma = GCN_{\sigma}\left(GCN(y^x), \mathbf{A}_{\sigma}, \ldots\right) \tag{37}$$

E. Time Complexity

The complete procedure of integrating VBPG into existing methods is outlined in Appendix B, available online. The computational cost of VBPG mainly lies in the reconstruction of gaussian distribution. The complete time complexity calculation is outlined in Appendix F, available online.

V. Experiments

A. Experiment Setup

(1) Datasets: We use Cora,[1] Citeseer[1], Pubmed[1], Blogcatalog[1] and ACM[1] as benchmark datasets to conduct our experiments. The details are shown in Appendix A, available online.

(2) Baselines: We implement the VBPG into various representative variational bayes-based methods to detect communities. These methods include end-to-end methods and k-means-based downstream methods. They are: VGAE [7], VGAER [8], RWR-VGAE [27], Sig-VAE [28], Graphite [9], WARGA [29], DG-VAE [30], VGAECD [31], GM-VGAE [11], CVGAE [23] and EG-VGAE&EGC-VGAE [12].

(3) Parameter setting and evaluation metrics: We integrate the VBPG plug-in into all baseline methods without modifying

[1] https://github.com/GDM-SCNU

TABLE I
COMMUNITY DETECTION RESULTS (%)

Methods	Cora			Citeseer			Pubmed			Blogcatalog			ACM		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
VGAE	47.98±0	67.49±0	43.84±0	19.23±1	43.62±2	16.35±1	18.38±0	61.28±0	18.16±0	24.77±0	41.30±0	16.65±0	47.73±4	80.25±4	49.31±6
+VBPG	**50.4±0**	**70.85±0**	**46.34±0**	**19.86±1**	**44.51±2**	**17.09±1**	**19.44±0**	**61.77±0**	**18.89±0**	**25.31±0**	**42.70±0**	**17.43±0**	**50.60±0**	**83.07±0**	**55.20±0**
VGAER	45.72±1	62.97±1	36.20±1	20.41±0	40.01±0	10.29±0	**29.71±0**	64.36±0	27.09±0	21.79±1	40.61±1	14.63±1	15.63±0	45.34±0	4.03±0
+VBPG	**46.26±1**	**63.73±1**	**36.98±1**	**20.57±0**	**40.09±0**	**10.31±0**	**29.71±0**	**64.37±0**	**27.10±0**	21.38±1	40.19±1	14.17±1	**15.70±0**	**45.44±0**	**4.05±0**
RWR-VGAE	46.21±0	56.25±0	37.38±0	25.27±1	44.54±1	17.83±0	30.22±0	67.42±0	29.81±0	25.26±0	44.17±0	15.63±0	35.95±0	67.74±0	29.35±0
+VBPG	**46.47±0**	**56.55±0**	37.06±0	**25.7±1**	**44.79±1**	**17.94±0**	**30.43±0**	67.33±0	**29.86±0**	**29.77±0**	43.16±0	**20.12±0**	**46.12±0**	**79.27±0**	**46.30±0**
Sig-VAE	48.58±2	64.00±4	39.08±4	24.48±0	45.20±2	18.02±1	25.93±0	66.62±0	25.23±0	12.35±0	30.59±0	6.25±0	9.94±0	48.56±0	10.72±0
+VBPG	**50.50±1**	**67.12±2**	**41.85±1**	**24.89±2**	**48.70±3**	**19.46±0**	**26.73±0**	65.95±0	**25.64±1**	**18.47±0**	**35.95±0**	**11.37±0**	**31.92±0**	**70.94±0**	**33.73±0**
Graphite	46.26±0	62.56±0	38.59±0	21.63±1	42.25±1	15.63±1	23.90±0	65.68±0	24.36±0	24.82±0	40.08±0	13.99±0	40.62±3	72.76±4	36.50±7
+VBPG	**47.46±0**	**64.11±0**	**39.12±0**	**22.93±1**	**44.20±2**	**19.05±2**	-	-	-	**24.93±0**	**40.11±0**	**14.59±0**	**44.88±0**	**80.94±0**	**50.86±0**
WARGA	47.03±0	62.86±0	39.11±0	31.23±0	55.60±0	27.67±0	6.45±0	44.38±0	6.24±0	24.90±0	39.36±0	14.28±0	30.70±4	64.50±5	20.71±5
+VBPG	**49.43±0**	**64.42±0**	**39.85±0**	**31.17±0**	**56.22±0**	**28.20±0**	**6.55±0**	**46.40±0**	**6.94±0**	24.53±0	**40.90±0**	**17.00±0**	**33.88±0**	**68.07±0**	**24.97±0**
DGVAE	44.62±3	58.43±6	34.72±6	18.05±1	41.55±2	13.36±2	24.16±0	65.08±0	23.71±0	25.23±0	39.82±0	15.22±0	35.87±0	68.95±0	26.47±0
+VBPG	**47.42±3**	**64.32±3**	**38.75±4**	**18.50±1**	**42.25±3**	**13.84±2**	-	-	-	**28.08±0**	**43.72±0**	**19.58±0**	**37.12±0**	**72.49±0**	**32.78±1**
VGAECD	49.46	67.61	47.50	38.92	61.14	37.00	4.46	46.62	4.83	19.98	35.54	8.28	53.85	84.49	60.18
+VBPG	**51.28**	**68.39**	47.19	**39.14**	**61.55**	**37.64**	**4.61**	**46.67**	**4.96**	**21.09**	**36.07**	**12.29**	**54.73**	**85.61**	**61.67**
GM-VGAE	44.54	55.06	32.83	15.15	35.98	13.01	19.49	61.20	17.73	29.86	49.11	22.81	26.77	59.57	25.13
+VBPG	**47.54**	**63.07**	**39.95**	**19.91**	**42.38**	**15.37**	**22.45**	**62.02**	**19.43**	**30.08**	48.36	**24.18**	**37.64**	**75.17**	**41.70**
CVGAE	51.14	71.31	47.80	31.63	59.84	31.92	29.43	68.26	29.90	27.19	46.50	19.35	58.37	86.64	64.13
+VBPG	**55.16**	**73.60**	**51.92**	**33.44**	**61.08**	**33.49**	**31.51**	**69.52**	**31.9**	**27.54**	46.15	18.82	**58.91**	**87.11**	**65.34**
EG-VGAE	53.20	**72.78**	50.45	41.97	68.65	43.95	29.45	71.86	32.76	25.86	36.97	11.99	65.23	88.76	68.95
+VBPG	**54.10**	**72.78**	**50.52**	**42.69**	**69.17**	**44.77**	**31.37**	**71.94**	**35.85**	**27.99**	**38.14**	**14.99**	**66.44**	**89.04**	**69.89**
EGC-VGAE	52.11	71.53	48.19	41.97	68.74	44.09	30.91	72.73	34.13	26.20	34.91	13.06	65.56	88.48	68.37
+VBPG	**52.47**	**71.60**	**48.19**	**42.49**	**68.98**	**44.56**	29.77	71.74	32.13	**27.21**	**38.78**	**16.24**	**66.26**	**89.41**	**70.66**
Avg.	1.80↑	2.31↑	1.84 ↑	0.95 ↑	1.40↑	1.05↑	0.68↑	0.24 ↑	0.57↑	1.51↑	1.27↑	2.39↑	4.83↑	5.88↑	7.78↑

(Bold: best; '-': out-of-memory.)

their original modules. For the VGAE utilizing the VBPG, we carefully tune the α and β within VBPG. Specifically, we set the α and β combination to (0.9, 0.6), (0.7, 0.9), (0.2, 0.8), (0.3, 0.1) and (0.7, 0.1) for the Cora, Citeseer, Pubmed, Blogcatalog and ACM datasets, respectively. Conversely, for the other baselines employing the VBPG, we directly set α and β to a constant of 1in all cases. All baseline methods either adopt the parameter settings described in their papers or, if they use different datasets, fine-tune them based on existing settings. For the update period $\mathcal{T}$, we consider that all baseline methods modulate the signal at least 4 times during training. All experiments are conducted on a PC running 64-bit Windows 10, equipped with a 3.40 GHz Intel Core i7-6700 CPU and 64 GB of RAM. The VBPG is implemented using Python 3.6, with additional dependencies listed in the code repository referenced in the abstract.

For all methods employing a downstream manner, we run k-means 10 times and report the mean and variance of the performance. For methods utilizing an end-to-end manner, we report the best performance. Furthermore, we employ three widely used metrics to measure the community detection performance of all methods, which are: NMI, Accuracy (ACC) and Adjusted Rand Index (ARI).

B. Community Detection Performance

The community detection results are reported in Table I. We have the following observations and analysis:

(1) Comprehensive experiments demonstrate the effectiveness of the VBPG: Compared to all baseline methods, these methods after integrating the VBPG generally show performance improvements. Specifically, by reconstructing the gaussian distribution via VBPG, these methods exhibit reasonable average improvements in NMI, ACC and ARI, with respective improvements of 4.83%, 5.88% and 7.78% on ACM. Notably, Sig-VAE, RWR-VGAE and GM-VGAE all demonstrate significant performance improvements. For example, within Sig-VAE, competitive improvements are observed across several metrics on the ACM dataset: NMI, ACC and ARI have increased by 21.98%, 22.38% and 23.01%, respectively. Furthermore, observations reveal that baseline methods incorporating VBPG exhibit similar average performance improvements across the Cora, Citeseer, and Blogcatalog datasets. Overall, NMI, ACC and ARI show improvements of 1.42%, 1.66% and 1.76%, respectively. Although experiments on the Pubmed dataset reveal that some methods (e.g., VGAER) using VBPG while ignoring parameters α and β only observe subtle performance changes, the adoption of VBPG generally shows promising performance improvements. For instance, in GM-VGAE, improvements in NMI, ACC and ARI are observed, with respective increases of 2.96%, 0.82% and 1.7%. In terms of performance stability, methods that use k-means for community detection tend to show better stability after integrating the VBPG plug-in. For example, on the ACM dataset, VGAE, Graphite, and WARGA all exhibited smaller performance deviations (i.e., smaller variance) compared to their original methods.

(2) Setting parameters for VBPG does not require concern in most cases: In our experiments, for all methods that incorporate VBPG, we fine-tune parameters α and β only in VGAE. For other methods, the α and β are set to 1. We can see that VGAE, when combined with VBPG and fine-tuning α and β, outperforms its original version on all datasets. Specifically, for NMI, ACC and ARI, it realizes substantial enhancements of 2.87%, 2.82% and 5.89% respectively on the Pubmed dataset. Meanwhile, sub-optimal results are observed on the Cora dataset. For the remaining methods, we can observe that, after integrating the

VBPG, variational bayes-based methods generally outperform their original method in most cases. For example, comparing performances before and after integrating the plug-in, the outcome reveals that both EG-VGAE and DGVAE experienced certain improvements. Therefore, considering the parameter sensitivity analysis presented in Section V-D, we conclude that variational bayes-based methods typically surpass their original versions after integrating the VBPG, regardless of whether α and β are considered. Although the observed performance improvements may not always be optimal, the non-necessity of parameter tuning simplifies the integration process of VBPG with existing methods and enhances its generality.

(3) VBPG effectively supports integration in both end-to-end and downstream manners: We are separately counting the performance improvements of methods applied in an end-to-end manner and those used in a downstream manner after integrating VBPG, and we find that both are yielding roughly similar results. Specifically, methods that adopt an end-to-end manner generally demonstrate slightly higher performance improvements on the Cora, Citeseer and Pubmed datasets compared to those employing a downstream manner. Among these, on the Citeseer dataset, the differences in NMI, ACC and ARI between the two types of methods are 1.13%, 0.62% and 0.21%, respectively. Conversely, on the ACM dataset, methods utilizing a downstream manner exhibit more pronounced differences of 3.41%, 3.77% and 5.61%, respectively. Overall, VBPG is applicable for two different community detection scenarios.

Analysis and discussion: For existing methods that integrate VBPG, desirable performance improvements are observed on most datasets. We attribute these improvements to several key factors: First, rules derived from our empirical study help mitigate the impact of community distortion and ambiguity on community detection results. Second, signal modulation avoids the loss of network potential information, a claim that is supported by experimental results presented in this section. This indicates that the reconstructed gaussian distribution fits the true prior distribution better than the original ones. From a theoretical perspective, the signal modulation strategy enhances the fitting accuracy of posterior distribution within the variational Bayes framework, bringing it closer to the true prior distribution. This theoretical advancement offers a novel approach to reconstructing the posterior distribution, surpassing the limitations of traditional methods.

Although certain metrics for some methods incorporating the VBPG show slight disadvantages in specific datasets, we are confident these can be offset by fine-tuning the parameters of VBPG. We think the possible reasons for this result are as follows: Some networks exhibit complex and irregular structures that may lead to differences between signal modulation and community characteristics within these networks. For example, in complex network structures like Pubmed, VBPG may struggle to fully capture the subtle relationships between nodes, leading to discrepancies between the signal modulation results and the actual community characteristics. Therefore, further research could focus on optimizing the signal modulation strategy for complex and irregular network structures. Additionally, the performance of methods integrating VBPG may be affected by the inherent limitations of their original methods. In particular, methods like EGC-VGAE, which are sensitive to structural changes in networks, may face performance challenges when undergoing signal modulation. This is because VBPG relies on modulating signal frequencies based on existing structural assumptions, which may not always align with the original design of the method. Therefore, future research could consider introducing relaxed constraints to allow VBPG to mitigate the impact of the inherent limitations of the original methods.

C. Ablation Experiments

We conduct ablation experiments on most methods to evaluate the impact of modulating signals solely during expectation inference or variance inference on community detection results. The experimental results are shown in Table II. We find that *simultaneously modulating signals during expectation and variance inference is optimal strategy, though it may not always yield the best community detection results.* We verify the conclusion through the analysis of two aspects:

(1) For some methods, it is necessary to simultaneously modulate the signals in expectation and variance inference. For example, we can observe the performance difference of Sig-VGAE on the ACM dataset with complete integration of VBPG versus only partial integration. Specifically, compared to complete integration of VBPG, modulating the signal only during variance inference results in a significant performance disadvantages, with differences of 17.89%, 17.99% and 18.87% for NMI, ACC and ARI, respectively. A similar observation can also be made in the GM-VGAE on the ACM dataset. Moreover, modulating the signal only during expectation inference can also negatively impact performance. Compared to the optimal results from complete integration of VBPG, the differences in NMI, ACC and ARI are 12.66%, 20.59% and 17.95%, respectively. Experimental results from methods such as VGAE and Graphite indicate that complete integration of VBPG generally outperforms partial integrations in most cases.

(2) Modulating the frequency signal only during expectation inference or variance inference is not generalizable. We observe that integrating only parts of VBPG can lead to the best performance results for some methods. For example, on the Cora dataset, modulating only the frequency signal of WARGA during variance inference results in improvements of 0.29% in NMI, 0.59% in ACC and 1.27% in ARI compared to complete integration of VBPG. Similarly, for Sig-VAE, modulating the frequency signal exclusively during expectation inference yields increases of 0.2% in NMI, 0.9% in ACC and 0.77% in ARI. A possible reason for our analysis is that different networks may have different structural features. For example, social networks may exhibit strong clustering features (i.e., holistic features), while information networks might rely more on specific attributes of nodes (i.e., node-specific features). As mentioned earlier, expectation inference mainly focuses on the central trend of the data, reflecting the overall changing trend of nodes in the network. Variance inference focuses on the volatility of data, reflecting the individual differences of nodes. Consequently, in certain networks, modulating the frequency signal solely during

TABLE II
ABLATION EXPERIMENT RESULTS (%)

Methods	Cora			Citeseer			Pubmed			Blogcatalog			ACM		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
VGAE+VBPG	**50.40**	**70.85**	**46.34**	**19.86**	**44.51**	**17.09**	**19.44**	**61.77**	**18.89**	**25.31**	**42.70**	**17.43**	**50.6**	**83.07**	**55.20**
$+\mu$	48.46	67.58	43.40	19.74	43.98	16.72	17.69	60.72	17.42	25.26	42.44	17.41	50.28	83.07	55.15
$+\sigma$	48.67	68.49	45.24	18.15	42.25	15.28	19.39	61.39	18.65	24.95	41.74	16.80	48.66	81.12	50.56
RWR-VGAE+VBPG	46.47	56.55	37.06	**25.70**	**44.79**	**17.94**	30.43	67.33	29.86	**29.77**	**43.16**	**20.12**	46.12	79.27	46.30
$+\mu$	**46.62**	**56.67**	36.89	24.05	43.83	17.44	29.96	67.18	29.49	29.71	42.65	20.10	**46.63**	**80.11**	**48.28**
$+\sigma$	46.30	56.44	**37.71**	25.04	44.24	17.10	**30.65**	**67.68**	**30.24**	25.43	42.24	15.81	36.23	67.77	29.43
Sig-VAE+VBPG	50.50	67.12	41.85	24.89	**48.70**	19.46	26.73	65.95	25.64	**18.47**	**35.95**	**11.37**	**31.92**	**70.94**	**33.73**
$+\mu$	**50.70**	**68.02**	**42.62**	**25.57**	47.50	**20.21**	26.14	65.50	24.96	18.23	35.92	11.21	23.00	65.48	24.43
$+\sigma$	47.51	61.58	37.04	24.40	44.39	16.83	**28.57**	**69.00**	**30.49**	12.40	30.43	6.19	14.03	52.95	14.86
Graphite+VBPG	**47.46**	**64.11**	**39.12**	**22.93**	**44.20**	**19.05**	-	-	-	**24.93**	**40.11**	**14.59**	**44.88**	**80.94**	**50.86**
$+\mu$	47.31	61.26	39.01	22.63	44.09	17.81	-	-	-	23.57	39.91	12.04	44.83	80.92	50.86
$+\sigma$	46.35	62.46	38.61	22.20	42.71	16.40	**-**	**-**	**-**	24.93	39.93	14.24	40.54	72.72	36.38
WARGA+VBPG	49.43	64.42	39.85	**31.17**	**56.22**	**28.20**	6.55	**46.40**	**6.94**	24.53	40.90	**17.00**	**33.88**	**68.07**	**24.97**
$+\mu$	47.14	62.97	38.65	31.01	56.07	28.19	6.33	45.92	6.42	26.03	**40.95**	15.03	33.62	67.68	24.38
$+\sigma$	**49.72**	**65.01**	**41.12**	31.12	55.74	27.68	**6.77**	45.10	6.84	**26.09**	39.60	14.35	33.37	67.16	23.72
DGVAE+VBPG	**47.42**	64.32	38.75	**18.50**	**42.25**	**13.84**	**-**	**-**	**-**	**28.08**	**43.72**	**19.58**	**37.12**	**72.49**	**32.78**
$+\mu$	47.13	64.56	**39.09**	17.66	40.45	13.58	-	-	-	27.38	43.31	19.08	36.77	71.80	31.46
$+\sigma$	46.67	**64.58**	38.77	18.13	41.94	13.74	**-**	**-**	**-**	26.09	43.13	16.61	35.97	69.17	26.84
VGAECD+VBPG	**51.28**	**68.39**	47.19	**39.14**	**61.55**	**37.64**	**4.61**	**46.67**	**4.96**	21.09	36.07	12.29	**54.73**	**85.61**	**61.67**
$+\mu$	51.14	68.13	47.62	34.09	60.08	32.72	4.21	44.56	3.78	21.22	39.65	14.99	53.85	84.99	60.18
$+\sigma$	51.27	68.06	**47.66**	34.04	60.41	33.02	3.83	44.08	3.52	**21.23**	**40.49**	**15.33**	53.85	84.99	60.18
GM-VGAE+VBPG	**47.54**	**63.07**	**39.95**	**19.91**	42.38	15.37	**22.45**	62.02	19.43	**30.08**	**48.36**	**24.18**	**37.64**	**75.17**	**41.7**
$+\mu$	46.61	55.80	36.52	19.89	44.94	19.33	16.18	57.23	14.27	19.75	38.88	13.65	24.98	54.58	23.75
$+\sigma$	44.37	61.30	37.06	24.37	**48.51**	**23.28**	22.28	**62.59**	**19.84**	23.36	39.92	14.15	30.13	62.35	25.40
CVGAE+VBPG	**55.16**	**73.60**	**51.92**	33.44	61.08	33.49	**31.51**	**69.52**	**31.90**	27.54	46.15	18.82	**58.91**	**87.11**	**65.34**
$+\mu$	51.98	71.90	48.48	31.56	59.75	31.63	29.97	68.68	30.51	27.36	46.50	19.29	58.69	86.84	64.69
σ	54.92	73.26	51.48	**34.46**	**61.71**	**34.77**	31.35	69.16	31.61	**28.13**	**47.67**	**19.42**	58.37	86.64	64.13
EG-VGAE+VBPG	**54.10**	72.78	50.52	**42.69**	**69.17**	**44.77**	31.37	71.94	35.85	**27.99**	**38.14**	**14.99**	**66.44**	**89.04**	**69.89**
$+\mu$	53.48	72.56	50.07	42.65	69.12	44.72	**31.99**	**73.76**	**35.85**	27.62	37.84	13.59	65.80	88.72	69.12
$+\sigma$	53.66	**72.93**	**51.01**	42.13	68.77	44.16	29.67	72.09	32.61	26.21	37.95	12.21	65.86	88.66	68.79
EGC-VGAE+VBPG	52.47	71.60	48.19	**42.49**	**68.98**	**44.56**	29.77	71.74	32.13	**27.21**	**38.78**	**16.24**	**66.26**	**89.41**	**70.66**
$+\mu$	52.49	71.64	48.53	42.35	68.86	44.37	**32.04**	**73.46**	**35.45**	27.14	36.86	15.48	65.70	88.71	68.88
$+\sigma$	**53.35**	**71.97**	**49.05**	41.89	68.71	44.05	29.50	71.65	32.00	26.42	35.68	14.85	65.75	89.04	69.79

($+\mu$: VBPG only modulates signals during expectation inference; $+\sigma$: VBPG only modulates Signals during variance inference.)

either expectation or variance inference could align more closely with the inherent characteristics of networks than modulating both simultaneously. Meanwhile, some methods that introduce additional constraints or integrate other modules into the vanilla variational bayes framework may exhibit different sensitivities to signal modulation. Although integrating only parts of VBPG may yield satisfactory results in specific contexts, it lacks broad applicability. The performance improvements from such partial integration tend to be limited compared to those achieved through complete integration of VBPG. This is observed when applying VBPG in RWR-VGAE on the Pubmed dataset, in EGC-VGAE on the Cora dataset, and in CVGAE on the Citeseer dataset.

Notably, while complete integration of VBPG may not always be optimal, methods employing this plug-in still maintain competitive performance compared to their original versions. This avoids the need to selectively modulate frequency signals under complex and uncertain conditions, ensuring broad generalizability of VBPG.

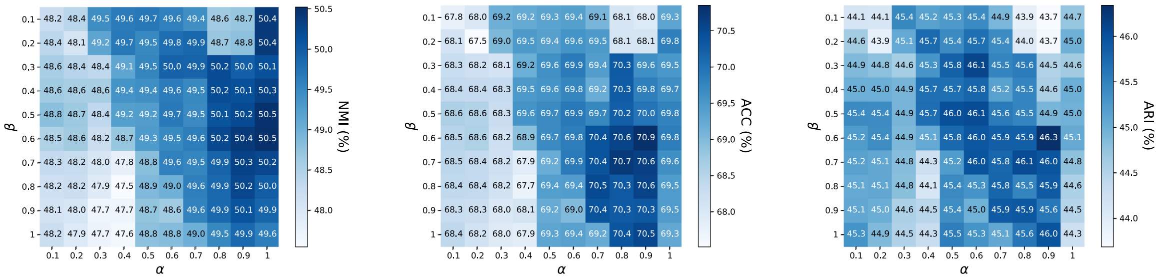
D. Sensitivity Analysis

Taking VGAE as a case study, we conduct a sensitivity analysis on the Cora dataset to evaluate how different VBPG parameters affect the performance of VGAE after integrating the plug-in. For α and β, we restrict their values to a range between 0.1 and 1. The experimental results are depicted in Fig. 8. Based on these results, we draw the following observations and analysis:

Overall, the performance impact of different parameter combinations is controllable, with variations in NMI, ACC and ARI staying within about 2.8%, 3.4% and 2.6%, respectively. Specifically, we observe better performance improvements when $\alpha \in [0.7, 0.9]$ and $\beta \in [0.6, 0.8]$. Noteworthy, keeping β constant, a decrease in α is associated with a downtrend in overall performance of VGAE. We propose the following possible explanations:

A higher α generally favors signal modulation during expectation inference, enabling a more effective capture overall data patterns and similarities among nodes. Conversely, a lower α might cause the low-frequency component to fail in reflecting important community characteristics, thus highlighting the negative impact of community distortion on the method performance. β mainly contributes to the differences features of nodes during variance inference. Choosing an appropriate β can highlight the differences among communities, thereby mitigating the negative impact of community ambiguity on method performance. Furthermore, we believe that the impact of different parameter combinations on the performance of methods integrating VBPG may also be connected to the inherent characteristics of the method itself, as well as to the complexity and irregularity of network characteristics.

Nonetheless, as demonstrated in our previous experimental results, even though ignoring α and β might not be the optimal



(a) NMI

(b) ACC

(c) ARI

Fig. 8. Sensitivity analysis: VGAE performance on Cora dataset under different α and β.

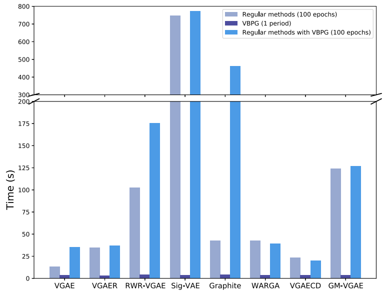


Fig. 9. Efficiency statistics.

strategy, competitive performance improvements can still be realized in most cases.

E. Efficiency Study

We conduct efficiency statistics on some methods. Specifically, we separately measure the time taken for 100 epochs of VGAE, VGAER, RWR-VGAE and other methods on the Cora dataset, the time taken by one period of VBPG, and the time taken for 100 updates of these method after integrating VBPG (note that all VBPG time overhead is deducted). The statistical results are shown in Fig. 9. Obviously, the time cost of VBPG is negligible, i.e., approximately 3 seconds per period. VBPG demonstrates its reusability as it acts as a plug-in to improve the performance of methods at a low time cost. For most methods, integrating VBPG results in a training time cost that is comparable to the original methods. Only a few methods show significant changes. We believe that the main reason is that the network structure after signal modulation is denser than the original topology. Consequently, the complexity of these methods and their limitations in handling non-sparse topologies necessitate longer training times after integrating VBPG. However, we observe that after integrating VBPG, most methods still maintain the same order of magnitude training time cost as the original methods. This demonstrates that integrating VBPG does not significantly increase the training cost of existing methods in most cases.

At the design level, VBPG is developed as a modular plug-in, demonstrating its flexible applicability in variational Bayes-based community detection methods. Experimental results show that VBPG can be seamlessly integrated into existing models, and in most cases, the training time after integration remains within the same order of magnitude as the original methods.

F. Robustness Study

We conduct robustness experiments and find that modulating frequency signals enhances the robustness of community detection methods to a certain extent. Detailed experimental results are given in Appendix C, available online.

G. Visualization

To intuitively illustrate the effectiveness of the VBPG in handling community distortion and community ambiguity, we conduct a visualized experiment. Detailed experimental results are given in Appendix D, available online. In general, integrating VBPG into existing methods alleviates the adverse effects of community distortion and ambiguity on community detection results.

H. Generalization Study

We employ the link prediction task to evaluate the generalization ability of VBPG. Detailed experimental results and analysis are given in Appendix E, available online. From an application perspective, although VBPG is initially designed for community detection and does not specifically account for the challenges of link prediction, its signal modulation approach demonstrates generalization ability in tasks like link prediction.

VI. Related Works

In this section, we briefly review related work on community detection in regular networks based on variational bayes methods. The challenge for methods based on variational bayes lies in improving inference and learning of the posterior distribution to approximate the unknown prior distribution. As mentioned before, existing methods can be broadly categorized into two types: downstream methods and end-to-end methods.

(1) Downstream methods: As a classic variational graph generation method, VGAE learns latent node representations through inference and generation under the constraints of ELBO. Owing to its competitive performance, interpretability and scalability, VGAE has gradually become an important method for

community detection. In recent years, numerous related variants have been developed to address specific challenges in graph representation and community detection, providing substantial insights into graph variational Bayes methods. Zhu et al. [32] proposes to learn the Gaussian distribution in Wasserstein space as the potential representation for each node. This approach contributes by not only preserving the network structure but also modeling node uncertainties, making it effective for real-world graphs with irregular structures. On this foundation, Liang et al. [29] directly regularizes the latent distribution of node embeddings to a target distribution by using the Wasserstein metric. Their work significantly enhances the quality of node representations by minimizing distortions in embedding space, especially in sparse networks. Additional, Vaibhav et al. [27] employs random walk-based regularization on the node representation z, effectively enhancing its ability to learn latent network information.

Unlike methods that directly impose constraints on the node representation in inference component, some methods also introduce higher-order network information or complex distributions in the generation model. Specifically, NetVAE [33] utilizes a shared encoder to jointly compress network infomation for transfer learning and information integration. Concurrently, it ntroduces a novel dual decoder architecture that leverages a Gaussian mixture model and block model to improve the quality of learned embeddings. Hasanzadeh el al. [28] proposes a hierarchical variational framework Sig-VGAE to better capture complex node dependencies. Their approach differs from traditional methods by using a Bernoulli-Poisson link decoder, which is especially effective in dealing with network sparsity, thereby improving the robustness of model on sparse and noisy graphs. Li et al. [30] propose establishing a connection between the generation term and balanced graph cuts, offering a principled approach to interpret the reconstruction terms as spectrally relaxed balanced graph cuts. This contribution provides a clearer understanding of how variational inference can be linked to graph partitioning objectives. The VGAER, proposed by Qiu et al. [8], introduces modularity information during the inference component and focuses on reconstructing this information rather than the network structure in the generation model to enhance community detection results.

(2) End-to-end methods: VGAECD, proposed by Choong et al. [10], integrates a Gaussian mixture model into the graph variational Bayes framework for end-to-end community detection. This work contributes by introducing a new ELBO formulation that directly constrains the inference and generation processes, ensuring that node representations effectively capture community memberships. Building on this, Choong et al. [34] introduces a simplified graph convolution encoder to accelerate convergence speed of VGAECD. By adopting dual variational objectives, their method demonstrates improved performance in both clustering accuracy and computational efficiency.

Based on the contributions of the above two works, many end-to-end community detection methods combined with Gaussian mixture models gradually emerge. For example, GM-VGAE [11] introduces a gaussian mixture model and an adversarial regularization to model the prior distribution in the variational graph bayes framework. FT-VGAE [35] is designed to solve the feature twist problem. By employing a novel edge decoding strategy that utilizes both neighborhood-level and cluster-level information, FT-VGAE improves clustering performance by mitigating the negative effects of feature twist. Mrabah et al. [23] further extend this by formulating a new ELBO that addresses differences between inference and generation modules. This modification helps avoid posterior collapse and mitigates feature drift, significantly improving the stability and accuracy of method. Cheng et al. [12] propose two novel methods, EG-VGAE and EGC-VGAE, which incorporate evolution information into static network analysis for community detection. EG-VGAE combines evolution information with static network information to enable fine-grained propagation of local low-frequency signals to global ones. EGC-VGAE is a variant of EG-VGAE that imposes smoothing constraints on adjacent slices of the network to reduce the impact of network noise. In addition, many works, including [36], [37], [38], [39], have contributed to community detection in an end-to-end manner. These works extend the application of variational Bayes methods in different domains, further demonstrating the versatility and effectiveness of these methods in a variety of network structures.

VII. Conclusion

In this paper, we revisit the application of variational bayes in community detection from a graph signal perspective and conduct an empirical study. This study allows us to summarize and generalize our findings. Specifically, compressing low-frequency signals during expectation inference and amplifying high-frequency signals during variance inference effectively mitigate the negative impact of community distortion and ambiguity on the results of community detection. Based on this insight, we propose a novel plug-in VBPG, to more accurately approximate the unknown prior distribution by reconstructing the gaussian distribution. Comprehensive experiments reveal that methods incorporating VBPG exhibit significant performance improvements. This work represents a breakthrough effort to combine signal processing with variational bayes theory.

In the future, VBPG is expected to be progressively applied in more complex scenarios, such as community detection methods based on variational bayes in dynamic networks. In addition, the performance of VBPG may be influenced by parameter selection, such as the balance between low- and high-frequency signals. While most methods may benefit from careful tuning, this can incur the additional overhead of increased human time costs. Therefore, introducing mechanisms to automate parameter selection could be a promising direction for future research.

References

[1] Y. Zhang, Y. Liu, Q. Li, R. Jin, and C. Wen, “LILPA: A label importance based label propagation algorithm for community detection with application to core drug discovery,” *Neurocomputing*, vol. 413, pp. 107–133, 2020.

[2] G. Siemens and R. S. J. de Baker, “Learning analytics and educational data mining: Towards communication and collaboration,” in *Proc. 2nd Int. Conf. Learn. Analytics Knowl.*, 2012, pp. 252–254.

[3] H. Zhang, P. Li, R. Zhang, and X. Li, “Embedding graph auto-encoder for graph clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9352–9362, Nov. 2023.

[4] L. Yang, Y. Wang, J. Gu, C. Wang, X. Cao, and Y. Guo, “JANE: Jointly adversarial network embedding,” in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 1381–1387.

[5] W. Xia, Q. Wang, Q. Gao, M. Yang, and X. Gao, “Self-consistent contrastive attributed graph clustering with pseudo-label prompt,” *IEEE Trans. Multimedia*, vol. 25, pp. 6665–6677, 2023.

[6] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proc. 2nd Int. Conf. Learn. Representations*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[7] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” 2016, *arXiv:1611.07308*. [Online]. Available: http://arxiv.org/abs/1611.07308

[8] C. Qiu, Z. Huang, W. Xu, and H. Li, “VGAER: Graph neural network reconstruction based community detection,” in *Proc. Int. Workshop Deep Learn. Graphs Methods Appl.*, 2022, pp. 1–9.

[9] A. Grover, A. Zweig, and S. Ermon, “Graphite: Iterative generative modeling of graphs,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 2434–2444. [Online]. Available: http://proceedings.mlr.press/v97/grover19a.html

[10] J. J. Choong, X. Liu, and T. Murata, “Learning community structure with variational autoencoder,” in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 69–78.

[11] G. Niknam, S. Molaei, H. Zare, D. Clifton, and S. Pan, “Graph representation learning based on deep generative Gaussian mixture models,” *Neurocomputing*, vol. 523, pp. 157–169, 2023.

[12] J. Cheng, C. He, K. Han, G. Chen, W. Liang, and Y. Tang, “Unveiling community structures in static networks through graph variational Bayes with evolution information,” *Neurocomputing*, vol. 576, 2024, Art. no. 127349.

[13] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, “Simplifying graph convolutional networks,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.

[14] X. Xie, W. Chen, Z. Kang, and C. Peng, “Contrastive graph clustering with adaptive filter,” *Expert Syst. Appl.*, vol. 219, 2023, Art. no. 119645.

[15] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Beyond low-pass filtering: Graph convolutional networks with automatic filtering,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6687–6697, Jul. 2023.

[16] D. Bo, X. Wang, C. Shi, and H. Shen, “Beyond low-frequency information in graph convolutional networks,” in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 3950–3957.

[17] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[18] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[19] E. Isufi, F. Gama, D. I. Shuman, and S. Segarra, “Graph filters for signal processing and machine learning on graphs,” *IEEE Trans. Signal Process.*, vol. 72, pp. 4745–4781, 2024.

[20] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” in *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[21] N. Liu, X. Wang, D. Bo, C. Shi, and J. Pei, “Revisiting graph contrastive learning from the perspective of graph spectrum,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, Art. no. 215.

[22] L. Hu, K. C. C. Chan, X. Yuan, and S. Xiong, “A variational Bayesian framework for cluster analysis in a complex network,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 11, pp. 2115–2128, Nov. 2020.

[23] N. Mrabah, M. Bouguessa, and R. Ksantini, “A contrastive variational graph auto-encoder for node clustering,” *Pattern Recognit.*, vol. 149, 2024, Art. no. 110209.

[24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. 5th Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[25] G. Peyré and M. Cuturi, “Computational optimal transport,” *Found. Trends Mach. Learn.*, vol. 11, no. 5/6, pp. 355–607, 2019.

[26] X. Zhang, H. Liu, Q. Li, and X. Wu, “Attributed graph clustering via adaptive graph convolution,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4327–4333.

[27] Vaibhav, P.-Y. B. Huang, and R. E. Frederking, “RWR-GAE: Random walk regularization for graph auto encoders,” 2019, *arXiv: 1908.04003*. [Online]. Available: https://api.semanticscholar.org/CorpusID:199543352

[28] A. Hasanzadeh, E. Hajiramezanali, K. R. Narayanan, N. Duffield, M. Zhou, and X. Qian, “Semi-implicit graph variational auto-encoders,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10711–10722.

[29] H. Liang and J. Gao, “Wasserstein adversarially regularized graph autoencoder,” *Neurocomputing*, vol. 541, 2023, Art. no. 126235. [Online]. Available: https://doi.org/10.1016/j.neucom.2023.126235

[30] J. Li et al., “Dirichlet graph variational autoencoder,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 443.

[31] J. J. Choong, X. Liu, and T. Murata, “Learning community structure with variational autoencoder,” in *Proc. 2018 IEEE Int. Conf. Data Mining*, 2018, pp. 69–78.

[32] D. Zhu, P. Cui, D. Wang, and W. Zhu, “Deep variational network embedding in Wasserstein space,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2827–2836.

[33] D. Jin, B. Li, P. Jiao, D. He, and W. Zhang, “Network-specific variational auto-encoder for embedding in attribute networks,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2663–2669.

[34] J. J. Choong, X. Liu, and T. Murata, “Optimizing variational graph autoencoder for community detection,” in *Proc. 2019 IEEE Int. Conf. Big Data*, 2019, pp. 5353–5358.

[35] N. Mrabah, M. Bouguessa, and R. Ksantini, “Escaping feature twist: A variational graph auto-encoder for node clustering,” in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 3351–3357.

[36] L. Guo and Q. Dai, “Graph clustering via variational graph embedding,” *Pattern Recognit.*, vol. 122, 2022, Art. no. 108334.

[37] R. Fei, Y. Wan, B. Hu, A. Li, and Q. Li, “A novel network core structure extraction algorithm utilized variational autoencoder for community detection,” *Expert Syst. Appl.*, vol. 222, 2023, Art. no. 119775.

[38] N. Mrabah, M. Bouguessa, and R. Ksantini, “Beyond the evidence lower bound: Dual variational graph auto-encoders for node clustering,” in *Proc. 2023 SIAM Int. Conf. Data Mining*, 2023, pp. 100–108.

[39] C. He, J. Cheng, Q. Guan, X. Fei, H. Li, and Y. Tang, “A deep conditional generative approach for constrained community detection,” in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 3928–3932.

[40] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Junwei Cheng received the BS degree from the School of Computer Science, University of Electronic Science and Technology of China, Zhongshan Institute, in 2020. He is currently working toward the PhD degree with the School of Computer Science, South China Normal University, China. His research interests include graph data mining.

Yong Tang received the BS and MS degrees from Wuhan University, China, in 1985 and 1990, respectively, and the PhD degree from the University of Science and Technology of China, China, in 2001, all in computer science. He is currently a professor with the School of Computer Science, South China Normal University, China. His research interests include Big Data and collaborative computing. He has published more than 100 papers on international journals and conferences.

Chaobo He received the BS, MS, and PhD degrees from South China Normal University, China, in 2004, 2007, and 2014, respectively. He is currently a professor with the School of Computer Science, South China Normal University, and is also a visiting scholar with the School of Data and Computer Science, Sun Yat-sen University, China. His research interests include machine learning and social computing. He has published more than 20 papers on international journals and conferences.

Pengxing Feng received the BEng degree in automotive engineering from China Agricultural University, in 2020, and the master's degree in automotive engineering from Tongji University, in 2023. He is currently working toward the PhD degree in electrical engineering with the City University of Hong Kong, Hong Kong SAR, China. His main research areas include statistical signal processing, and adaptive signal processing.

Kunlin Han received the BS degree from the School of Computer Science, South China Normal University, China, in 2021. He is currently working toward the MSc degree with the Department of Computer Science, University of Southern California, United States. His research interests include machine learning and computer architecture.

Quanlong Guan (Member, IEEE) received the MS and PhD degrees from Jinan University, China, in 2006 and 2014, respectively. He is currently a professor with the Faculty of Computer Science and the deputy dean with the Research Institute for Guandong Intelligent Education, Jinan University, Guangzhou, China. His research interests include smart education, the application of artificial intelligence, system reliability and security, data protection, and safety.