Task-Oriented Communication for Graph Data: A Graph Information Bottleneck Approach

Shujing Li, Yanhu Wang, Shuaishuai Guo, *Senior Member, IEEE*, and Chenyuan Feng, *Member, IEEE*

Abstract—Graph data, essential in fields like knowledge representation and social networks, often involves large networks with many nodes and edges. Transmitting these graphs can be highly inefficient due to their size and redundancy for specific tasks. This paper introduces a method to extract a smaller, task-focused subgraph that maintains key information while reducing communication overhead. Our approach utilizes graph neural networks (GNNs) and the graph information bottleneck (GIB) principle to create a compact, informative, and robust graph representation suitable for transmission. The challenge lies in the irregular structure of graph data, making GIB optimization complex. We address this by deriving a tractable variational upper bound for the objective function. Additionally, we propose the VQ-GIB mechanism, integrating vector quantization (VQ) to convert subgraph representations into a discrete codebook sequence, compatible with existing digital communication systems. Our experiments show that this GIB-based method significantly lowers communication costs while preserving essential task-related information. The approach demonstrates robust performance across various communication channels, suitable for both continuous and discrete systems.

Index Terms—Task-oriented communication, graph neural network, graph information bottleneck, vector quantization.

I. INTRODUCTION

IN the swiftly changing realm of communication systems, marked by the emergence of 5G and its advancements, an increasing demand arises for efficient and smart communication frameworks capable of adjusting to intricate and ever-shifting environments. Conventional communication systems concentrate on enhancing metrics such as throughput and latency, frequently neglecting the distinct needs of task-oriented communications. In task-oriented scenarios, the objective extends beyond the mere effective transmission of raw data; it aims to optimize the performance for specific tasks like classification, prediction, or decision-making. It is worth noting that in addition to Euclidean data such as images and text, there are vast amounts of non-Euclidean data generated in social networks, biological networks, transportation networks and recommendation systems, etc. [1] This graph-structured data has no order or coordinate reference points, and is difficult to represent in a grid-like matrix or a tensor. A graph is usually composed of nodes and edges, which contain rich relational information. We can regard the graph data as a

Shujing Li, Yanhu Wang and Shuaishuai Guo are with School of Control Science and Engineering, Shandong University, Jinan 250061, China and also with Shandong Key Laboratory of Wireless Communication Technologies, Shandong University, Jinan, Shandong, P. R. China (e-mail: lishujing777@mail.sdu.edu.cn, yh-wang@mail.sdu.edu.cn, shuaishuai_guo@mail.sdu.edu.cn).

Chenyuan Feng is with Department of Communication Systems, EURECOM, Sophia Antipolis 06410, France (e-mail: Chenyuan.Feng@eurecom.fr).

web of relationships, wherein the nodes are the subjects and the edges represent the relationships between the nodes. This feature makes graph an effective tool for modeling complex relationships in various non-Euclidean data scenarios [2]–[6].

Graph data is widely applied to knowledge representation, recommendation systems, and user behavior analysis [7]. However, the inherent complexity and size of graph data can pose challenges for transmission and storage. A complete graph, which contains all possible relationships between the nodes, is often not suitable for direct transmission due to the substantial bandwidth and storage resources required. Moreover, a complete graph tends to contain redundant information, leading to unnecessary resource consumption. In many practical applications, only specific task-related information within the graph needs to be transmitted to accomplish particular objectives. Additionally, graph data might contain sensitive information or personal privacy that should be safeguarded during transmission. Transmitting the entire graph increases the risk of information leakage. Therefore, an intelligent and concise communication system is needed to transmit graph data effectively. Within this context, the Graph Information Bottleneck (GIB) stands out as an innovative strategy to tackle these challenges. It draws upon the principles of information theory and graph neural networks to craft solutions. Central to this approach is the compression of transmitted information, ensuring that it retains only the most pertinent features relevant to the task. This targeted compression facilitates a more efficient and impactful mode of communication.

A. Related Works

Task-oriented communication is a promising solution for graph data transmission. Different from traditional communication, it focuses on the accurate transmission of task-relevant information, rather than the bit-level precise transmission [8], [9]. Its characteristics align well with our demands for graph data transmission. It is difficult to characterize or extract task-related information with mathematical models, and the existing task-oriented communication systems mainly rely on deep learning technology [10]. That is, the neural networks (NNs)-based encoder and decoder are constructed and trained, so that the task-oriented communication system obtains the ability to extract task-relevant information [11]–[13].

Precisely, Farsad et al. proposed a neural network architecture for text transmission task, that combines joint source-channel coding (JSCC) with recurrent neural network (RNN) based encoder, binarization layers, channel layers, and RNN-based decoder [14]. Xie et al. introduced a system for textual semantic restoration tasks based on Transformer model

[9]. Their approach focuses on maximizing system capacity while minimizing semantic errors by restoring the meaning of sentences. Guo et al. proposed to utilize pre-trained language models to quantify the semantic importance of text and allocate unequal power based on semantic importance [15]. Some JSCC methods map image pixel values directly to input representations to achieve high-quality image reconstruction tasks [16], [17]. Kang et al. proposed an image transmission method specifically designed for scene classification tasks [18]. Shao et al. developed a task-oriented communication scheme for edge inference using the information bottleneck (IB) [11], [13]. Their scheme aims to improve the performance of image classification task at the edge server by efficiently transmitting relevant information. These NNs-based task-oriented communication systems have achieved outstanding outcomes, demonstrating the capacity to effectively execute specific tasks.

B. Motivation & Contributions

However, these works cannot be directly used for graph data transmission. Because they are based on CNNs or fully connected networks and deal with regular data such as text (represented as sequences of characters) and images (represented as continuous two-dimensional or three-dimensional pixel sets). Graph data exhibits a more intricate structure, which is composed of nodes and edges with attribute information. When transmitting graph data, it is important to consider the efficiency of transmitting large-scale graph data while accurately preserving the correlation between nodes and edges [19], [20]. Therefore, it is necessary to design a new task-oriented communication system for graph data transmission. Using information bottleneck theory to develop task-oriented communication systems is a good option [11], [21]. Because IB theory aims to find an optimal intermediate representation that can retain important information in the input data to ensure the accuracy of the output prediction while eliminating redundant information [22], [23]. However, it should be noted that the direct utilization of IB-based frameworks in graph data processing is not feasible. This is because the IB framework assumes that the data follows an independent identically distributed (IID) pattern. Whereas in graph data, the presence of edges and their attributes results in the data points being dependent on one another, which makes the graph data deviate from the IID assumption [24], [25].

Recently, an information-theoretical design principle for graph data, named GIB, has been developed, which seeks the right balance between data compression and information preservation for graph data [26]. Specifically, employing GNNs[1] as the foundational framework for graph data processing, GIB works well. Through the process of learning and

---

[1]GNN is a deep learning model specifically designed to work with graph data. In contrast to traditional deep learning models primarily handling vectorized data, GNNs excel at capturing intricate relationships within graphs, involving nodes and edges [27]. The fundamental principle of GNNs is to gradually aggregate local neighborhood information by iteratively updating the representation of nodes [28]. Typical GNN models include Graph Convolutional Networks (GCNs), Graph Isomorphism Network (GIN), Graph Attention Networks (GATs), etc.
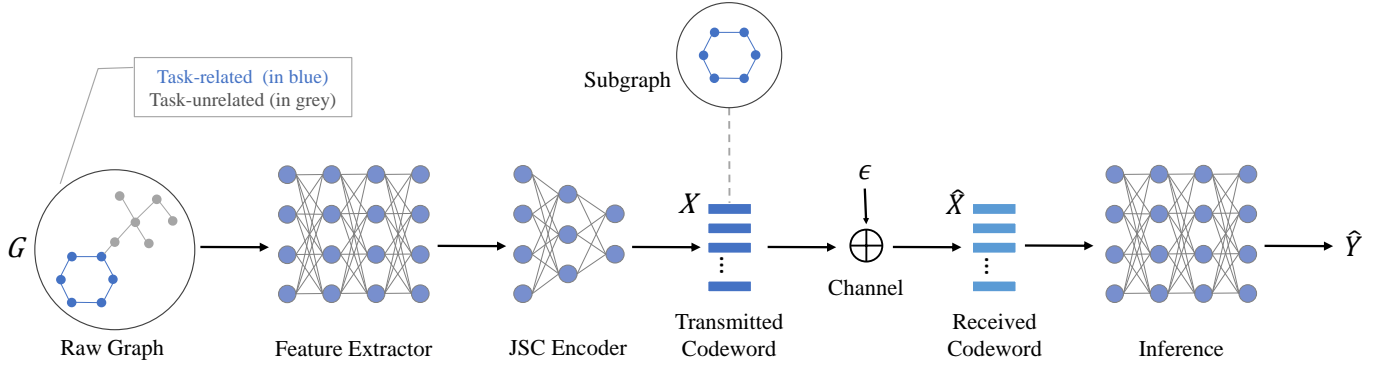
parameter adjustments, GIB ensures the preservation of only task-relevant information while simultaneously compressing extraneous data. One of the challenges in studying graph information bottleneck, i.e., how to handle the interdependencies between nodes in graph data, is the non-IID character of graph data. One of the primary technical contributions of our research is task-oriented transmission, which we will further optimize for.

In addition, the actual widespread use of digital communication systems makes the realization of compatibility between task-oriented communication systems and digital communication systems a necessity. Thus, there is a requirement to identify a suitable digitization mechanism for communication systems geared towards graph data, ensuring both effective data compression and robustness. Vector quantization (VQ) emerges as a powerful technique that addresses these needs by mapping high-dimensional data into a finite set of lower-dimensional codewords. This process not only facilitates significant data compression but also enhances the system's robustness against noise and transmission errors. Furthermore, by integrating with deep learning, VQ is highly adaptive and can be dynamically adjusted to different data distributions and channel conditions.

Motivated by these issues, we design GIB-enabled task-oriented communication systems for graph data in this work. Our main contributions are summarized as follows:

- We introduce GIB into a task-oriented communication system for graph data. We build a Markov chain model for information transmission and formulate an optimization problem to maximize the mutual information between the task target and received codewords while minimizing the mutual information between the received codewords and the raw graph data. This approach balances the preservation of critical information in the extracted features while eliminating redundant information, thereby enhancing task success rates and reducing communication overhead.
- To address the difficulty of handling mutual information terms in GIB due to high-dimensional integration, we use the Mutual Information Neural Estimator (MINE) to directly estimate the mutual information between the original graph and the subgraphs. This approach overcomes the challenge of obtaining the prior distribution of the subgraphs in applying variational approximation methods.
- Recognizing the importance of topological information in graph data, particularly in revealing community structures, we introduce a connectivity loss term into the objective function. This term leverages topological information during feature extraction, reduces fluctuations in ambiguous node assignments, and contributes to a more stable training process.
- We map the resulting subgraph representation onto a jointly trained codebook to generate a discrete index sequence for transmission. This mapping ensures compatibility with existing digital communication systems. Experimental results show that the system is able to achieve higher compression rates while achieving task

Fig. 1. Task-Oriented Communication Scheme for Graph Data: In this scheme, the "transmitted codeword "refers to the encoded representation of the task-related subgraph, while the "corrupted codeword "denotes the representation of the subgraph received by the receiver, which has undergone corruption during transmission through the channel. The symbol $\hat{Y}$ represents the task inference output.

success rates comparable to traditional digital communication methods.

C. Organization

The subsequent sections of this paper are organized as follows. In Section II, we outline the system model, describe the structure and design objectives of the task-oriented communication system for graph data. Section III introduces the details of GIB-enabled task-oriented communication systems for graph data, including the handling of GIB objectives and the system training strategy. Section IV presents our proposed approach for digitizing the task-oriented communication system. In Section V, we evaluate the performance and effectiveness of the proposed method through experiments. In Section VI, potential applications of the proposed method in practical scenarios are discussed. Finally, we make a brief summary of this paper in Section VII.

D. Notations

In this paper, a graph with m nodes is defined as g or $G = (V, E, A, \mathcal{X})$, where $V = \{V_i | i = 1, 2, ..., m\}$ is the set of nodes with cardinality m, $E = \{(V_i, V_j) | i < j, \ V_i \ and \ V_j \ are \ connected\}$ is the edge set, $A = \{0, 1\}^{m \times m}$ is the adjacent matrix, and $\mathcal{X} \in R^{m \times d}$ is the feature matrix corresponding to V with feature dimension d. The pair (G, Y) stands for the graph data and its target variable. The entropy of Y is defined as $H(Y)$. The mutual information between X and Y is represented as $I(X, Y)$.

II. SYSTEM MODEL AND PROBLEM DESCRIPTION

A. System Model

In this paper, we consider a task-oriented communication system for graph data as shown in Fig. 1. The system mainly includes a transmitter and a receiver, where the transmitter consists of a feature extractor implemented by a GNN and a joint source-channel (JSC) encoder. The receiver is a neural network used for task inference. Given an input graph g, they cooperate to perform tasks to make the inference output $\hat{y}$ consistent with the true target variable y. The random variables (G, Y) together with the encoded codeword X and the channel-corrupted codeword $\hat{X}$ form the following probabilistic model:

$$(Y) \, G \to X \to \hat{X} \to \hat{Y}, \tag{1}$$

which satisfy $p(\hat{y}|g) = p_\theta(\hat{y}|\hat{x}) \, p_{channel}(\hat{x}|x) \, p_\phi(x|g)$. Herein, the lowercase letters g, x, $\hat{x}$, y, and $\hat{y}$ denote realizations of the variables represented by the corresponding uppercase letters. $p_\phi(x|g)$ represents the transmitter neural network parameterized by adjustable parameters ϕ. For a given input graph g, the feature encoder identifies the representation x related to the task. The task-relevant features are encoded by the JSC encoder and then transmitted to the receiver through the channel.

To accommodate both continuous channels and discrete channels within this framework, we introduce an indicator variable ζ that distinguishes between the channel types: $\zeta = 0$ for an continuous channel and $\zeta = 1$ for an discrete. The conditional probability distribution $p_{channel}(\hat{x}|x)$ is thus defined as:

$$p_{\text{channel}}(\hat{x}|x; \zeta) = \begin{cases} \frac{1}{\sqrt{2\pi N_0}} \exp\left(-\frac{(\hat{x}-x)^2}{2N_0}\right), & \text{if } \zeta = 0 \\ P_{\hat{X}|X}(\hat{x}|x). & \text{if } \zeta = 1 \end{cases} \tag{2}$$

where we assume that the continuous channel is the additive white Gaussian noise (AWGN) channel and the discrete channel is the Symmetric Discrete Channel (SDC).

AWGN is implemented using an untrained neural network layer, and the transfer function is expressed as:

$$\hat{x} = x + \epsilon, \tag{3}$$

where the Gaussian noise $\epsilon \sim \mathcal{N}(0, N_0/2)$.

For SDC, $P_{\hat{X}|X}(\hat{x}|x)$ signifies the transition probability matrix, encapsulating the likelihood of transitioning from input to output symbols. The SDC model assumes that both channel inputs and outputs utilize the same symbol set, with each symbol corresponding to an identical output probability distribution. This probabilistic behavior is succinctly captured

by a transition matrix:

$$\mathcal{P} = \begin{bmatrix} \varepsilon & \frac{1-\varepsilon}{r-1} & \cdots & \frac{1-\varepsilon}{r-1} \\ \frac{1-\varepsilon}{r-1} & \varepsilon & \cdots & \frac{1-\varepsilon}{r-1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1-\varepsilon}{r-1} & \frac{1-\varepsilon}{r-1} & \cdots & \varepsilon \end{bmatrix}_{r \times r}, \qquad (4)$$

where ε represents the probability of correct symbol transmission and r denotes the cardinality of the channel symbol set. Potential transmission errors could lead to the receiver retrieving incorrect vectors, impacting the inference task.

The conditional distribution $p_\theta(\hat{y}|\hat{x})$ stands for the task inference network parameterized by adjustable parameters θ at the receiver. It deduces the class label $\hat{y}$ based on $\hat{x}$ based on the channel-corrupted codeword $\hat{x}$.

B. Problem Description

The effectiveness of task execution is intricately linked to the dimension of the feature vector produced by the transmitter. For instance, in the context of graph classification tasks, a higher dimension of the feature vector could lead to an elevated classification accuracy. However, this advantage comes at the cost of increased communication overhead. Thus, the challenge is to identify a concise and informative representation that aligns with the optimal subgraph. To formalize this trade-off, we formulate the following objective function based on the GIB principle:

$$\mathcal{L}_{GIB} = -I\left(Y, \hat{X}\right) + \beta I\left(G, \hat{X}\right), \qquad (5)$$

in which $I(Y, \hat{X})$ represents the mutual information capturing the relevance of task-specific information in the received codeword, $I(G, \hat{X})$ represents the preserved information in $\hat{X}$ given G, and β acts as a trade-off factor governing the relationship between the two.

Developing task-oriented communication based on GIB is a promising approach to solving the graph transmission challenges. This approach offers an information metric for the graph data and can effectively capture node and edge information. However, there are a few noteworthy problems:

- **Problem 1**: How to deal with mutual information containing graph data to get a tractable objective function? The expansion of the first term in (5) yields:

$$\begin{aligned} I\left(Y, \hat{X}\right) &= \int p(y, \hat{x}) \log \frac{p(y, \hat{x})}{p(y)p(\hat{x})} dy d\hat{x} \\ &= \underbrace{- \int p(y, \hat{x}) \log p(y) dy d\hat{x}}_{H(Y) = \text{constant}} \\ &\quad + \int p(y, \hat{x}) \log p(y \mid \hat{x}) dy d\hat{x}, \end{aligned} \qquad (6)$$

where the first term of the second equation is the entropy of Y. For a given graph and a given task, the entropy of Y is determined, so this term can be regarded as a constant and can be ignored in subsequent optimization.

Therefore, we can obtain the expression for (6) in the optimization implication:

$$I\left(Y, \hat{X}\right) = \int p(y, \hat{x}) \log p(y \mid \hat{x}) dy d\hat{x}. \qquad (7)$$

The joint distribution $p(g, y)$ for graph data and target labels is known. $p_\phi(\hat{x}|g)$ is determined by the transmitter network $p_\phi(x|g)$ and channel model $p_{channel}(\hat{x}|x; \epsilon)$. For the second integral term, we derive according to the Markov chain:

$$p(y \mid \hat{x}) = \int \frac{p(g, y)p_\phi(\hat{x} \mid g)}{p(\hat{x})} dg, \qquad (8)$$

Next, we address the second mutual information term of (5) and expand it:

$$I\left(G, \hat{X}\right) = \int p(\hat{x} \mid g)p(g) \log \frac{p_\phi(\hat{x} \mid g)}{p(\hat{x})} dg d\hat{x}, \qquad (9)$$

where $p(\hat{x})$ is also an intractable high-dimensional integral:

$$p(\hat{x}) = \int p(g)p_\phi(\hat{x} \mid g) dg. \qquad (10)$$

It is customary in IB to substitute this marginal distribution with a tractable prior distribution [29]. In GIB, finding a suitable variational prior is difficult. This is because of GIB's interpretation of $p(\hat{x})$: it represents the distribution of irregular subgraph structures, not just a latent graph data representation. Moreover, due to the non-IID nature of graph data, finding a simple function as a prior distribution is not feasible. Therefore, a new method is needed to estimate that mutual information.

- **Problem 2**: How to utilize topological information in graph data for communication tasks?
  By designing a feature extractor based on GIB theory, it becomes possible to selectively extract task relevant information, reducing redundancy and improving efficiency. Yet, when GNNs extract features from graph data, they often concentrate too much on nodes and neglect the graph's structural features. However, for many tasks, the crucial information is deeply rooted in the graph's topology. Therefore, in the process of feature extraction, it is necessary to consider the topological structure of the graph. To do this, we need to impose constraints on the feature extractor to improve the role of topological information in feature extraction.

- **Problem 3**: How to be compatible with digital communication systems?
  While the proposed task-oriented communication system effectively addresses the challenge of graph data transmission, its use of continuous signals poses compatibility issues with existing digital communication systems. Given the well-established infrastructure of digital communication systems, it is impractical to abandon it completely. Therefore, it is necessary to develop a task-oriented communication system compatible with the digital communication system to transmit graph data.

III. GIB-Enabled Task-oriented Communication

In this section, we develop a task-oriented communication system based on GIB theory. Specifically, we combine variational approximation and MINE techniques to improve the GIB formulation [30]. This modification enables a more streamlined optimization process employing neural networks, which solves **Problem 1**. In addition, we describe the process of feature extraction in detail and also node assignment, including the network configuration and the pertinent revision made to the objective function, which solves **Problem 2**.

A. Graph Information Bottleneck Reformulation

To solve **Problem 1**, we introduce the variable distribution $q_\theta(y|\hat{x})$ as a substitute for the true posterior distribution $p(y|\hat{x})$. θ represents the parameter of the inference neural network in the receiver computing the inference output $\hat{y}$. Consequently, (7) is transformed into:

$$I\left(Y,\hat{X}\right) = \int p(y,\hat{x}) \log q_\theta(y|\hat{x}) \, dy d\hat{x}$$
$$= E_{p(y,\hat{x})}\left[\log q_\theta(y|\hat{x})\right]. \tag{11}$$

Subsequently, Monte Carlo sampling is used to derive the empirical distribution of the joint distribution as an approximation. N samples are taken for the given data, that is:

$$p(y,\hat{x}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_y(y_i)\,\delta_{\hat{x}}(\hat{x}_i). \tag{12}$$

Here, $\delta(\cdot)$ denotes the Dirac function utilized for sampling the training data. y_i and $\hat{x}_i$ represent the label and representation received by the receiver for the i-th training data, respectively. This approximation leads to a tractable optimization objective from (11):

$$\mathcal{L}_{inf}(q_\theta(y\mid\hat{x})) = -\frac{1}{N}\sum_{i=1}^{N}\log q_\theta(y_i\mid\hat{x}_i). \tag{13}$$

Minimizing the objective function in (5) is consistent with maximizing $I\left(Y,\hat{X}\right)$ and minimizing $\mathcal{L}_{inf}$. This function is defined as the loss between the inference result Y and the ground truth for the received $\hat{x}$. A smaller loss signifies superior performance of the inference neural network.

About the second mutual information term $I\left(G,\hat{X}\right)$ of (5), indicating the transmission informativeness, we employ the MINE to directly approximate it without the need to estimate $p(\hat{x})$. We express this term in the form of Kullback-Leibler (KL)-divergence:

$$I\left(G,\hat{X}\right) = D_{KL}\left[P\left(G,\hat{X}\right)\|P(G)\otimes P\left(\hat{X}\right)\right]. \tag{14}$$

For ease of analysis and mathematical treatment, we adopt the Donsker-Varadhan representation of KL-divergence, which expresses the mutual information term as the difference between the expected value and the logarithmic expected value:
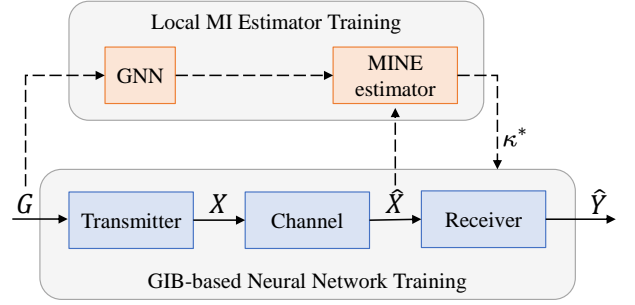

Fig. 2. Training Framework for GIB-Based Task-Oriented Communication Systems: The process begins with training the mutual information estimator to acquire a set of optimized MINE parameters, denoted as κ^*. Following this, the entire neural network undergoes training, during which the mutual information estimator utilizes the previously determined MINE parameters.

$$I\left(G,\hat{X}\right) = \sup_{T:G\times\hat{X}\to\mathbb{R}} E_{P(G,\hat{X})}[T] - \log E_{P(G)P(\hat{X})}\left[e^T\right]$$
$$= \mathcal{L}_{MI}(T), \tag{15}$$

in which $T = f_\kappa(g,\hat{x})$ encompasses all functions that render two expectations finite. The MINE can be used to estimate the mutual information between regular input data and its vector representation. Due to the irregularity of graph data, GNN is employed to extract the vector representation before feeding it into the multi-layer perceptron (MLP) for processing alongside the representation $\hat{X}$. $f_\kappa(\cdot)$ serves as the statistics network, denoting the neural network that executes the process of obtaining the corresponding real numbers from G and $\hat{X}$. The optimization of this estimator involves adjusting the MINE parameter κ such that the value on the right-hand side of (14) closely approximates $I\left(G,\hat{X}\right)$. This optimization is formalized as:

$$\max_\kappa \mathcal{L}_{MI}\left(\kappa,\hat{X}\right) = \frac{1}{K}\sum_{i=1}^{K}f_\kappa(g_i,\hat{x}_i) - \log\frac{1}{K}\sum_{i=1,j\neq i}^{K}e^{f_\kappa(g_i,\hat{x}_i)}. \tag{16}$$

After training with K sets of training data, the training process generates a set of suboptimal MINE parameters denoted as κ^* (see Fig. 2).

Drawing from (11) and (15), we derive a reformulation of the overall loss function:

$$\mathcal{L}_{GIB}(\phi,\theta,\kappa^*) = E_{p(g,y)}\Big\{E_{p_\phi(\hat{x}|g)}\left[-\log q_\theta(y|\hat{x})\right]$$
$$+ \beta\left[E_{p_\phi(\hat{x}|g)}f_{\kappa^*}(g,\hat{x}) - \log E_{p(\hat{x})}e^{f_{\kappa^*}(g,\hat{x})}\right]\Big\}. \tag{17}$$

By combining the sampling method, equations (13) and (16), we arrive at a tractable optimization problem for the entire system:

$$\min_{\phi,\theta}\mathcal{L}_{GIB}(\phi,\theta,\kappa^*) = \mathcal{L}_{inf}(\theta,\phi) + \beta\mathcal{L}_{MI}\left(\kappa^*,\hat{X}\right). \tag{18}$$

B. Model Training Strategy

To solve **Problem 2**, we specially design the feature extraction module. Node assignment is the main step in the graph feature extraction process. First, we introduce assignment probabilities as a continuous relaxation in node assignment, to address the problem that the discrete nature of the graph make it difficult to optimize (18) using gradient optimization methods. Next, we introduce connectivity loss to improve the impact of topological information on feature extraction, which can also make the node assignment process more stable. The detailed design process of this module will be discussed below.

We devise a node assignment mechanism to determine the inclusion of each node in the graph within the subgraph. This mechanism involves GNN extracting node features X from the original graph G. These features are then input into an MLP to obtain preliminary node assignments. The Softmax function is subsequently employed to convert these assignments into a probabilistic form. We represent the selected subgraph as G_{sub}. Specifically, for a given graph G, nodes either belong to G_{sub} or $\overline{G}_{sub}$. The node assignment mechanism yields a matrix S:

$$S = \text{Softmax}\left(MLP_{\sigma_2}\left(X\right)\right) \; with \; X = GNN_{\sigma_1}\left(A, X\right). \tag{19}$$

The matrix S has dimensions $m \times 2$, where m is the number of nodes in the input graph, and each row is a two-dimensional vector indicating the assignment probability of the corresponding node. Specifically, the i-th row of S,

$$\left[p\left(V_i \in G_{sub} \mid V_i\right), p\left(V_i \in \overline{G}_{sub} \mid V_i\right)\right],$$

represents the probability of the i-th node belonging to G_{sub} or $\overline{G}_{sub}$.

We define $\phi_1 = \{\sigma_1, \sigma_2\}$ as the parameter of the node assignment mechanism, which is part of the parameter of the transmitter. Once adequately trained, the values in each row of S should converge to 0 or 1, achieving robust node assignment. We take the first row of $S^T X$ to obtain the feature vectors of the n nodes belonging to the subgraph.

In conjunction with the JSC encoder parameter ϕ_2, we obtain the transmitter parameter $\phi = \{\phi_1, \phi_2\}$. For simplicity, we collectively refer to the node assignment mechanism and the JSC encoder as the feature extractor and encoder, denoted as $g_\phi(\cdot)$. Consequently, the objective function is explicitly expressed as:

$$\min_{\phi, \theta} \mathcal{L}_{GIB}\left(\phi, \theta, \kappa^*\right) = \mathcal{L}_{inf}\left(q_\theta\left(g_\phi\left(G\right)\right)\right) + \beta \mathcal{L}_{MI}\left(\kappa^*, \hat{X}\right). \tag{20}$$

We introduce a continuous relaxation with probabilistic assignment of nodes, alleviating the problems arising from the discreteness of the graph. Nevertheless, inadequate initialization may lead to poorly trained node allocation mechanisms and failure to achieve the desired outcomes. In other words, if the probabilities of a node belonging to G_{sub} or not are too close, nodes amay not be appropriately assigned. On one hand, over-assigning nodes to G_{sub} will lead to the presentation of a subgraph that includes an excessive amount of redundant information. On the other hand, assigning too few nodes to G_{sub} will result in an inadequate amount of task-

---

Algorithm 1 Model Training Procedure

1: **Input:** Graph and class label pairs $\{G, Y\}$, dimension of the encoder output D, batch size B, channel variance $\frac{N_0}{2}$

2: **while** epoch t=1 to T **do**
3:     Select a mini-batch of data $\{(g_b, y_b)\}_{b=1}^B$
4:     Extract node feature vectors
5:     Compute the node assignment matrix S
6:     Compute the encoded feature vectors of subgraphs X
7:     **while** batch b=1 to B **do**
8:         Sample noise ϵ N times for each pair of (g_b, y_b)
9:     **end while**
10:     Compute the connectivity loss based on (21)
11:     Compute the inference loss based on (13)
12:     **while** inner loop $k = 1$ to K **do**
13:         Tanin the MINE by (16) to get the suboptimal κ^*
14:     **end while**
15:     Compute the MI loss with κ^*:
$$\mathcal{L}_{MI} = \frac{1}{N}\sum_{i=1}^N f_{\kappa^*}\left(g_i, \hat{x}_i\right) - \log \frac{1}{N}\sum_{i=1, j \neq i}^N e^{f_{\kappa^*}\left(g_i, \hat{x}_i\right)}$$
16:     Compute the overall loss $\mathcal{L}_{GIB}$ based on (17)
17:     Calculate gradients and update model parameters by backpropagation
18: **end while**

---

related information in the subgraph, rendering it incapable of successfully executing the task.

To address the above issues, we assume that the model has an inductive bias that helps the model to focus more on the connectivity relationships between nodes. The model is thus better able to capture and utilize the topological information of the graph. We incorporate the connectivity loss proposed in [31] to introduce this inductive bias:

$$\mathcal{L}_{con} = \left\|Norm\left(S^T A S\right) - I_2\right\|_F, \tag{21}$$

where $Norm(\cdot)$ denotes row normalization, I_2 is the 2×2 identity matrix , and $\|\cdot\|_F$ is the Frobenius norm. Elements a_{11} and a_{12} in the first row of $S^T A S$ are defined as follows:

$$a_{11} = \sum_{i,j} A_{ij} p\left(V_i \in G_{sub} \mid V_i\right) p\left(V_j \in G_{sub} \mid V_j\right), \tag{22}$$

and

$$a_{12} = \sum_{i,j} A_{ij} p\left(V_i \in G_{sub} \mid N_i\right) p\left(V_j \in \overline{G}_{sub} \mid V_j\right). \tag{23}$$

Intuitively, if a node belongs to G_{sub}, its neighboring nodes are highly probable to belong to G_{sub} as well. Conversely, if a node does not belong to G_{sub}, its adjacent nodes likely do not belong to G_{sub} either. Consistently with this, ensuring adequate nodes are assigned to G_{sub} while reducing redundancy is achieved through $\frac{a_{11}}{a_{11}+a_{12}} \to 1$. This occurs simultaneously with reducing redundancy in G_{sub} through $\frac{a_{12}}{a_{11}+a_{12}} \to 0$. Analogously, this holds for $\overline{G}_{sub}$ and the elements of the second row of $S^T A S$.

To summarize, we introduce the continuous relaxation with probabilistic node assignment to enhance the optimization process. To address potential challenges stemming from inad-

equate initialization, such as ambiguous node assignment and unstable training process, we incorporate the inductive bias with the connectivity loss $\mathcal{L}_{con}$. The overall loss function is refined as:

$$\min_{\phi,\theta} \mathcal{L}_{GIB}\left(\phi,\theta,\kappa^*\right) = \mathcal{L}_{inf}\left(q_\theta\left(g_\phi\left(G\right)\right)\right)$$
$$+\beta\mathcal{L}_{MI}\left(\kappa^*,\hat{X}\right) + \alpha\mathcal{L}_{con}\left(g_\phi\left(G\right)\right). \quad (24)$$

The training procedures for the GIB-enabled task-oriented communication system are illustrated in Algorithm 1.

C. Computational Complexity Analysis

The computational complexity inherent in the proposed methodology primarily emanates from its reliance on GNNs. In our analysis, the GCN is chosen as the representative GNN to elucidate the computational complexity of the system. Characteristically, GCN employs a one-hop receptive field to assimilate local features, subsequently enlarging this receptive field through the stratification of layers. Denoting the number of GCN layers by L, the computational complexity attributed to GNNs can be articulated as $\mathcal{O}\left(|E|\sum_{i=1}^{L} D_{in}^{(i)} D_{out}^{(i)}\right)$, where $|E|$ signifies the quantity of edges in the graph, and $D_{in}^{(i)}$, $D_{out}^{(i)}$ represent the input and output dimensions of each respective layer. In parallel, the complexity associated with the MLP layer and the Softmax function within the node assignment mechanism is quantified as $\mathcal{O}(mD)$ and $\mathcal{O}(m)$ respectively, with m indicating the total number of graph nodes and D symbolizing the output dimension of GNNs. Consequently, the cumulative computational complexity of the proposed framework can be comprehensively expressed as $\mathcal{O}(|E|\sum_{i=1}^{L} D_{in}^{(i)} D_{out}^{(i)} + ND)$.

IV. Digitization of GIB-Enabled Task-Oriented Communications

This section extends the analog task-oriented communication system, introduced in Section III, to suit digital communication environments. As outlined in **Problem 3**, practical applications necessitate compatibility with digital systems. We address this by integrating vector quantization for digital transmission adaptation.

A. Digital Transmission

A pivotal strategy for transitioning to a digital framework involves discrete codebook mapping. This technique maps the continuous outputs of the neural network encoder into discrete codewords, as discussed in related works [32], [33]. The mapping process employs a predefined discrete codebook alongside nearest neighbor rules [34]. Let $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ be the continuous outputs from the encoder described in Section III. The codebook, denoted as $\mathcal{E} \in \mathbb{R}^{K \times d}$ consists of K codewords $e_1, e_2, \ldots, e_K \in \mathbb{R}^d$, where K represents the codebook size. The discrete mapping function is formulated as:

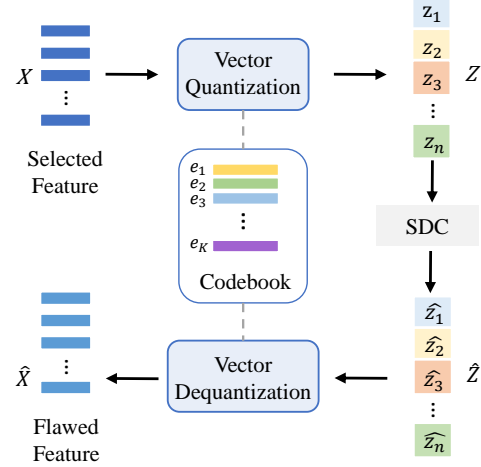$$x_{i\_q} = e_k \text{ where } k = \arg\min_j \|x_i - e_j\|_2. \quad (25)$$

Fig. 3. System Digitization Modules. This figure illustrates the process where the subgraph representation is matched against a codebook to generate an index sequence via the nearest neighbor principle. This sequence is sent through a symmetric discrete channel, subject to a specified error probability, leading to potential discrepancies between the transmitted and received index sequences. The receiver then reconstructs the representation vector using a shared codebook with the transmitter for subsequent task inference.

Employing discrete codebook mapping, the probabilistic model of the system is transformed into a new Markov chain, represented as:

$$(Y)\, G \to X \to Z \to \hat{Z} \to \hat{X} \to \hat{Y}, \quad (26)$$

which adheres to the following relationship:

$$p\left(\hat{y}|g\right) = p_\phi\left(x|g\right) p_Q\left(z|x\right) p_{SDC}\left(\hat{z}|z\right) p_{DQ}\left(\hat{x}|\hat{z}\right) p_{inf}\left(\hat{y}|\hat{x}\right). \quad (27)$$

Here, the transition from X to Z entails discretizing feature representations using the codebook. Z represents the index sequence derived via the nearest neighbor rule, substituting X as the input to the channel. For the discrete signal Z, we employ the discrete channel defined in (2) for its transmission, that is:

$$p_{SDC}\left(\hat{z}|z\right) = p_{\text{channel}}(\hat{z}|z; \zeta = 1) = P_{\hat{Z}|Z}(\hat{z}|z), \quad (28)$$

where $P_{\hat{Z}|Z}(\hat{z}|z)$ is the transition probability matrix of SDC described in Section II when $\zeta = 1$.

In digital transmission, as indicated in (25), each vector x_i from the encoder output is aligned with its nearest embedding vector e_k from the shared codebook. The transmitter's role is simplified to sending the index k to the receiver. The assignment of indices for transmission is defined as:

$$p(Z_i = k \mid x_i) = \begin{cases} 1 & \text{for } k = \arg\min_j \|x_i - e_j\|_2 \\ 0 & \text{otherwise} \end{cases}. \quad (29)$$

This results in a continuous vector being translated into a discrete one-hot codeword. However, channel imperfections can lead to erroneous index detection ($\hat{Z}$). We utilize the SDC

to model this transmission of discrete indices.

B. Design of Discrete Codebook

Each codeword e_i in the vector quantizer's codebook defines a Voronoi region $\mathcal{V}_i$:

$$\mathcal{V}_i = \left\{ x \in R^d : \|x - e_i\| \leq \|x - e_j\|, \text{ for all } j \neq i \right\}. \quad (30)$$

These regions collectively span the entire vector space $\mathbb{R}^d$ to which the encoder's output belongs. The quantization process entails finding the codeword closest to a given vector, which subsequently determines the Voronoi region and index for the vector. Balancing the size of the codebook is crucial as increasing it reduces quantization distortion but also elevates computational complexity.

We opted for a moderate-sized, suboptimally designed codebook to achieve acceptable quantization results. To enhance the codebook's performance, we introduce a learnable aspect, allowing codewords to adapt during training. Additionally, the encoder's output range is constrained to prevent extreme variations.

The non-differentiable nature of the quantization operation poses a challenge for encoder training. To address this, a straight-through estimator is employed, allowing gradients from the decoder input to flow back to the encoder output. Furthermore, a loss term ($\mathcal{L}_{vq}$) is added to reduce the distance between the encoder output and the corresponding codeword:

$$\mathcal{L}_{vq} = \|sg\,[x] - e\|_2, \quad (31)$$

with $sg\,[\cdot]$ representing a stop gradient function.

A commitment loss ($\mathcal{L}_{CM}$) is also introduced to ensure the encoder's output does not deviate excessively from the codewords:

$$\mathcal{L}_{CM} = \|x - sg\,[e]\|_2. \quad (32)$$

The final objective function ($\mathcal{L}_{VQ-GIB}$) encompasses these elements alongside the mutual information and connectivity losses, guiding the entire system:

$$\mathcal{L}_{VQ-GIB} = \mathcal{L}_{inf}\left(q_\theta\left(g_\phi\left(G\right)\right)\right) + \beta\mathcal{L}_{MI}\left(\kappa^*, \hat{X}\right) \\ + \alpha\mathcal{L}_{con}\left(g_\phi\left(G\right)\right) + \mathcal{L}_{vq} + \lambda\mathcal{L}_{CM} \quad (33)$$

subject to the optimized mutual information parameters κ^*:

$$\kappa^* = \arg\max_\kappa \mathcal{L}_{MI}\left(\kappa, \hat{X}\right). \quad (34)$$

$$\mathcal{L}_{vq} = \|sg\,[x] - e\|_2, \quad (35)$$

where $sg\,[\cdot]$ denotes stop gradient. Therefore, this item is solely valid for codebook learning, not encoder training.

Given that the volume of the encoder's output space is dimensionless, achieving satisfactory quantization mappings is challenging if the training of the codewords in the codebook does not align with the rate of parameter training in the encoder. Therefore a commitment loss is introduced to constrain the encoder's output from increasing:

$$\mathcal{L}_{CM} = \|x - sg\,[e]\|_2. \quad (36)$$

TABLE I: Architecture of Neural Network for GIB-Enabled Task-Oriented Communications

	Layer		Output Dimensions
Encoder Network	GCN Layer		(num, dim)
	GCN Layer		(num, dim)
	Node Assignment	Fully-connected Layer + Tanh	(num, dim)
		Fully-connected Layer + Softmax	(num, 2)
	Aggregation		(batch, dim)
Channel	AWGN Channel		(batch, dim)
Decoder Network	Fully-connected Layer + Dropout		(batch, dim)
	Fully-connected Layer + Log-softmax		(batch, class_n)

This item only affects the encoder, bringing the output closer to the code vector without influencing codebook learning.

For codebook updating, we adopt an exponential moving average (EMA) approach akin to K-means clustering. This method assigns variable weights to cluster centers over training batches, ensuring that the codebook dynamically adapts to the encoder's output.

V. EXPERIMENTS AND DISCUSSIONS

In this section, we evaluate the performance of the proposed GIB-based task-oriented communication system on graph classification tasks, and investigate the adaptability and effectiveness of continuous and discrete communication systems respectively. Additionally, ablation studies are also conducted to illustrate the contributions of the MI loss $\mathcal{L}_{MI}$ in GIB and the connectivity loss $\mathcal{L}_{con}$ presented in Section III, as well as the impact of the trade-off factor β on the system performance.

A. Experimental Setup

1) Datasets: For our graph classification experiments, we carefully select two datasets: COLLAB and PROTEINS.

- **COLLAB**: This scientific collaboration dataset represents a researcher's ego network, where nodes correspond to researchers and edges indicate collaboration between them. Each researcher's ego network is labeled based on the field to which the researcher belongs, resulting in three possible labels. COLLAB consists of 5,000 graphs, with an average of 74 nodes and 2,457 edges per graph.
- **PROTEINS**: This dataset comprises 1,113 proteins classified as enzymes or non-enzymes. Nodes in the graph represent amino acids, and edges exist between nodes if the distance between corresponding amino acids is less than 6 angstroms. On average, each graph in the PROTEINS dataset has 39 nodes and 73 edges.

2) Settings and Baselines: To assess the performance of the proposed method in a graph classification task, we integrate GIB into two distinct backbones: Graph Convolutional Network (GCN) [35] and Graph Isomorphism Network (GIN) [36].

We compare the proposed method with other graph-level representation learning methods: **InfoGraph** based on mean

TABLE II: Architecture of Neural Network for GIB-Enabled Discrete Task-Oriented Communications.

	Layer	Output Dimensions
Encoder Network	GCN Layer	(num, dim)
	GCN Layer	(num, dim)
	Node Assignment	(batch, dim)
	Aggregation	(batch, dim)
	Vector Quantization	(batch, size)
Channel	Symmetric Discrete Channel	(batch, size)
Decoder Network	Vector Decoding	(batch, dim)
	Fully-connected Layer + Dropout	(batch, dim)
	Fully-connected Layer + Log-softmax	(batch, class_n)

aggregation [35] and **ASAP** (Adaptive Structure Aware Pooling) based on pooling aggregation [37] in terms of graph classification accuracy, respectively.

- **InfoGraph**: InfoGraph is an unsupervised graph-level representation learning method that maximizes the mutual information between the representations of entire graphs and the representations of substructures at different granularity (e.g., nodes, edges, triangles) to make the graph representations adequately capture the features of substructures.
- **ASAP**: ASAP utilizes self-attention network along with GNNs to capture the importance of each node in a given graph. It also learns a sparse soft cluster assignment for nodes at each layer to pool the subgraphs to form the pooled graph.

For a fair comparison, all methods use the same number of GNN layers in the backbone. Models are trained using Stochastic Gradient Descent (SGD) with the Adam optimizer. We employ 10-fold cross-validation to report classification accuracy in experiments to validate the performance of the models.

3) Neural Network Architecture: While we do not enforce complete consistency in the network architectures of different approaches, we ensure that the GNN backbone and the number of layers are consistent across methods. The neural network architectures for the proposed method are shown in Tables I and II, which are outlined below. In the tables, num is the number of nodes, batch is the batch size during training, class_n is the number of graph classes, and dim represents the hidden dimension. The size in Table II refers to the size of the codebook, which was set to 256 in the experiment.

- Table I shows the network architecture of the proposed method where GCN is used as the backbone. We use two GCN layers to extract the node features of the graph and then perform node assignment. The output of the Node Assignment layer is multiplied with the full node features extracted by the GCN to obtain the node features of the selected subgraph. This output undergoes power normalization before passing through the AWGN channel.
- Table II illustrates the network architecture of the proposed digital communication system. In discrete communication systems, subgraph representations require quantization before entering the channel. During the training

process, the codebook utilized for vector quantization undergoes automatic updates until it reaches a stabilized state. The index sequence obtained from quantization passes through a symmetric discrete channel with a certain error transfer probability, and the decoder performs dequantization based on the codebook shared with the encoder.

We note that the hidden layer dimensions of the proposed method and baseline methods are set to the same. To comply with the original implementation idea of the baseline methods, the output dimension of their encoder network is twice as large as that of the proposed method, consuming more communication resources. In addition, since the vector quantization part is not involved in baselines, the same quantization method as the proposed method is adopted for comparison algorithms.

B. Performance of GIB-enabled Task-oriented Communications Without Vector Quantization

In the experiments, we evaluated the robustness of the proposed method amidst varying channel conditions. Prior to transmission, we implemented signal power normalization. During the training phase, a consistent SNR of 5 dB was maintained, whereas for testing, the SNR was methodically varied within a range extending from -15 dB to 25 dB. Two distinct experimental sets were executed, each with a batch size of 128. The first set operated with a hidden dimension of 16, while in the second, this parameter was augmented to 32.

Fig. 4 graphically illustrates the inference performance of the methods under evaluation across different channel quality scenarios. A discernible trend was noted, wherein the inference accuracy of all three methods demonstrated progressive enhancement in conjunction with rising SNR levels during testing, culminating in a plateau. Specifically, Fig. 4 (a) delineates the variation of classification accuracy relative to SNR for the three methods on the PROTEINS dataset. Our proposed GIB-based method, alongside the ASAP method, showcased robust classification performance, with the former exhibiting particularly notable efficacy. The InfoGraph method achieved commendable classification accuracy, especially under near-ideal channel conditions and with a higher hidden dimension. Fig. 4 (b) presents the experimental outcomes on the COLLAB dataset, a social network dataset characterized by complex topology and diverse node interrelations. The presence of densely connected communities within such networks significantly influences the graph's structural information, which in turn impacts the classification task. Both baseline methods exhibited suboptimal performance on this dataset, attributable to their inadequate consideration of graph topology. As explicated in Section III, the connectivity loss $\mathcal{L}_{con}$ steers the model towards heightened consideration of the graph's structural characteristics, enhancing the performance of our approach.

Fig. 5 compares the graph classification performance between our method, which employs the GIN as the backbone, and the alternative method utilizing the GCN. These experiments were conducted on both the PROTEINS and COLLAB datasets, with hidden dimensions set to 16 and 32, respectively. Notably, in the realm of graph classification, the GIN-based
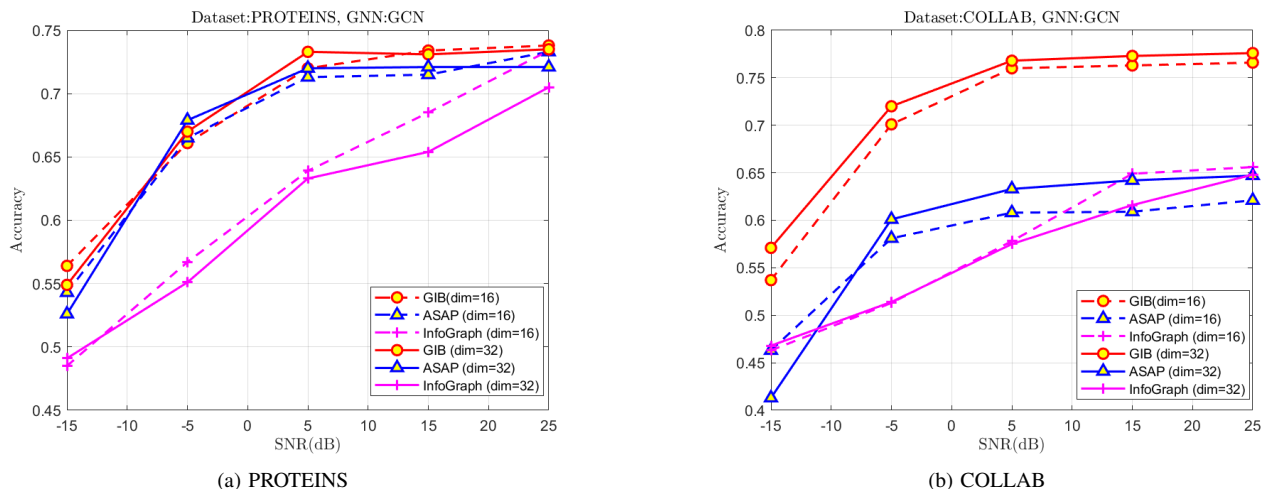
(a) PROTEINS

(b) COLLAB

Fig. 4. Classification accuracy variation with SNR for three methods on PROTEINS and COLLAB Datasets using GCN backbone, at hidden dimensions of 16 and 32.
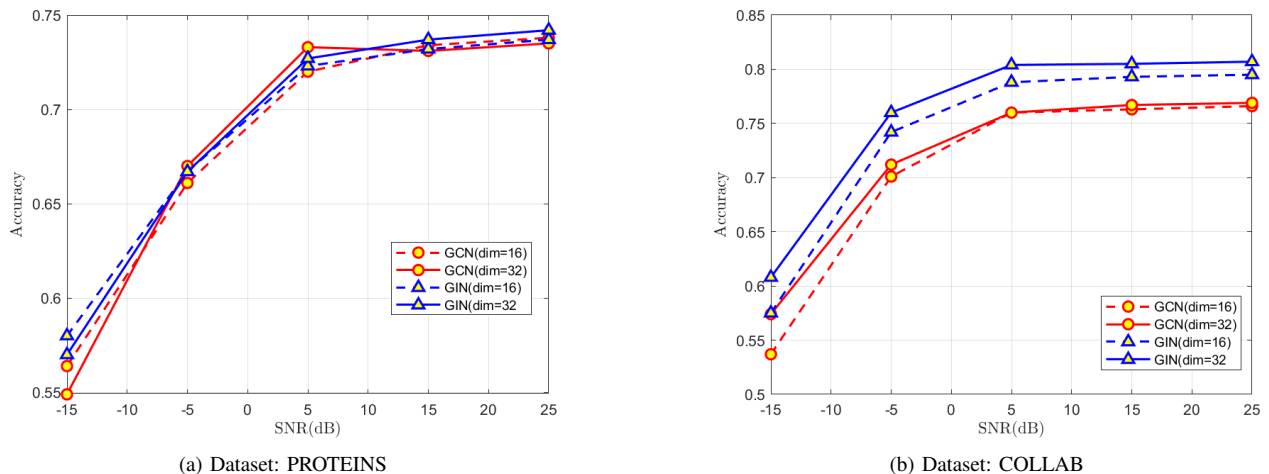

(a) Dataset: PROTEINS

(b) Dataset: COLLAB

Fig. 5. Classification accuracy variation with SNR for the proposed method on PROTEINS and COLLAB datasets, utilizing GCN and GIN backbones at hidden dimensions of 16 and 32.

method surpassed its GCN-based counterpart. The GCN, a prevalent choice for graph data processing, updates node representations through uniform aggregation of neighboring nodes. In contrast, GIN adopts a distinct methodology, where each node initially aggregates its feature with those of its neighbors via weighted summation. This non-commutative aggregation process, which is indifferent to the order of nodes, imparts to the GIN model an invariance to graph isomorphism, thus enabling it to effectively capture both the structure and global information inherent in the graph.

Given the specific focus on graph classification in this study, the adoption of GIN as the backbone is recommended. However, for a broader spectrum of tasks, the selection of GNNs with varying intrinsic characteristics as backbones can significantly bolster task-specific performance.

C. Performance of GIB-enabled Task-Oriented Communication Systems Utilizing Vector Quantization

In this part, the performance of the proposed discrete communication system, which incorporates digital codebooks,

is rigorously evaluated. Our experiments are structured to assess system robustness under varying probabilities of correct transmission. Here, the probability of correct transmission, denoted as ε, signifies the likelihood that the index received by the receiver accurately corresponds with that transmitted by the sender. Consequently, higher values of ε are indicative of superior channel quality. During the model training phase, ε is fixed at 0.94, whereas for testing, an array of ε values, specifically [0.90, 0.92, 0.94, 0.96, 0.98], are examined.

Fig. 6 graphically represents the classification performance of all three evaluated methods as a function of ε, with the hidden dimensions set at 16 and 32 respectively. The results delineated in these figures unequivocally demonstrate that our method significantly outperforms the baseline methods under various channel quality scenarios. The integration of InfoGraph with VQ is observed to be substantially ineffective for graph classification tasks within digital communication systems. The ASAP method exhibits moderate effectiveness on the PROTEINS dataset; however, its efficacy is markedly diminished when applied to the more complex COLLAB
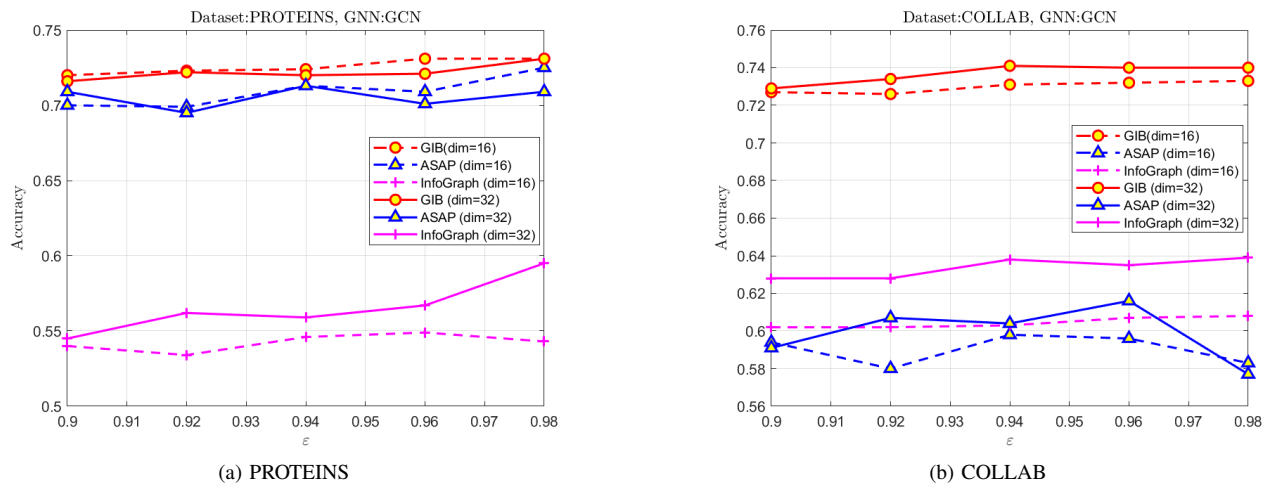
Fig. 6. Classification accuracy variation with probability of correct transmission (ε) for three methods on PROTEINS and COLLAB datasets using GCN backbone at hidden dimensions of 16 and 32.

dataset. As illustrated in Figure 6 (b), its classification accuracy is confined to approximately 0.6 or lower, indicating that minor transmission errors exert negligible impact on the outcomes. The curve illustrating the fluctuation of classification accuracy in relation to ε is characterized by irregular and unpredictable oscillations. In stark contrast, our proposed method demonstrates robust performance across both datasets, with classification accuracy exhibiting a modest but consistent upward trend in line with improvements in channel quality. This outcome aligns well with our initial hypotheses and theoretical underpinnings.

To further evaluate the performance of our proposed task-oriented digital communication system based on the GIB and VQ, we compared it against traditional digital communication method. Additionally, to validate the effectiveness of the VQ mechanism in task-oriented graph data transmission scenarios, we conducted experiment combining GIB with 8-bit scalar quantization followed by Quadrature Phase Shift Keying (QPSK) modulation. Given the non-uniformity in the dimensionality of the signals produced by these methods, we apply the same error rate per symbol to all methods to ensure that the assessment is based on the methods' ability to handle equivalent levels of channel-induced errors, rather than on their inherent dimensions. Models were trained with a fixed error rate of 0.01 and tested across a range of error rates from 0.06 to 0.014. These experiments were carried out on two datasets, PROTEINS and COLLAB, respectively. The experimental results are shown in Fig. 7.

On PROTEINS, all methods exhibited comparable outcomes. This is because the PROTEINS data set is relatively simple and does not require a more elaborate symbolic representation. However, on COLLAB, the performance of the GIB based methods is significantly better than that of traditional digital communication method, which reflects the effectiveness of GIB for extracting task-related information on complex graph data. Between the two GIB-based methods, vector quantization significantly outperforms the traditional digitization method, highlighting its advantages in handling complex datasets with higher dimensionality and complexity.

TABLE III: Impact of different hidden dimensions on the performance of GIB and GIB without $\mathcal{L}_{MI}$ on PROTEINS dataset.

SNR		-15 dB	-5 dB	5 dB	15 dB	25 dB
GIB	dim-16	0.564	0.661	0.720	0.734	0.738
	dim-32	0.549	0.670	0.733	0.731	0.735
	diff	**-0.015**	**0.009**	**0.013**	**-0.003**	**-0.003**
GIB without $\mathcal{L}_{MI}$	dim-16	0.544	0.634	0.717	0.728	0.733
	dim-32	0.535	0.640	0.730	0.731	0.734
	diff	**-0.009**	**0.006**	**0.013**	**0.003**	**0.001**

TABLE IV: Impact of different hidden dimensions on the performance of GIB and GIB without $\mathcal{L}_{MI}$ on COLLAB dataset.

SNR		-15 dB	-5 dB	5 dB	15 dB	25 dB
GIB	dim-16	0.559	0.725	0.762	0.767	0.765
	dim-32	0.586	0.723	0.770	0.774	0.778
	diff	**0.027**	**-0.002**	**0.008**	**0.007**	**0.013**
GIB without $\mathcal{L}_{MI}$	dim-16	0.537	0.701	0.758	0.763	0.765
	dim-32	0.571	0.720	0.769	0.773	0.776
	diff	**0.034**	**0.019**	**0.011**	**0.010**	**0.011**

The superior performance of the proposed task-oriented digital communication system based on the GIB and VQ on the COLLAB dataset highlights its potential advantages over traditional communication systems and traditional digital schemes, particularly in scenarios where data is characterized by high dimensionality and intricate patterns.

D. Ablation Study

To empirically ascertain the contribution of the individual components in our proposed methodology, we executed a series of ablation studies. Specifically, we extracted $\mathcal{L}_{MI}$ and $\mathcal{L}_{con}$ from the loss function, thereby deriving two distinct variants of our method: GIB lacking $\mathcal{L}_{MI}$ and GIB lacking $\mathcal{L}_{con}$.
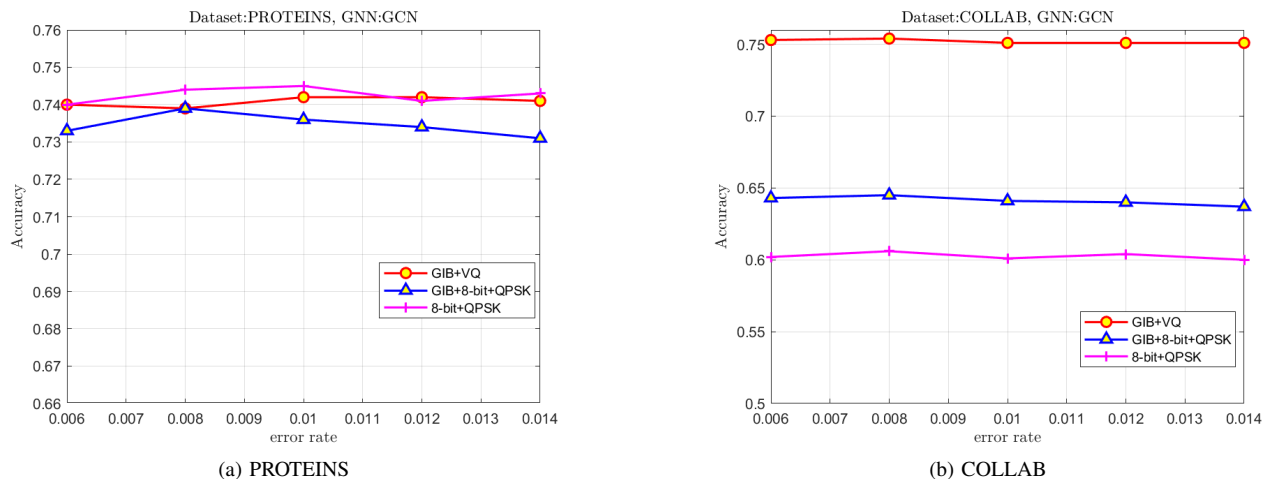
Fig. 7. Classification accuracy of three methods varying with the symbol error rate on the PROTEINS and COLLAB datasets: GIB combined with vector quantization, GIB combined with scalar quantization, and traditional digital communication using 8-bit + QPSK modulation.

TABLE V: Task accuracy and standard deviation of GIB and GIB without $\mathcal{L}_{con}$ on the PROTEINS dataset at the 16-dimensional hidden dimension.

SNR		-15 dB	-5 dB	5 dB	15 dB	25 dB
GIB	mean	0.564	0.661	0.72	0.734	0.738
	stdev	0.049	0.050	0.036	0.028	0.027
GIB without $\mathcal{L}_{con}$	mean	0.532	0.648	0.724	0.737	0.725
	stdev	0.052	0.058	0.039	0.033	0.030

TABLE VI: Task accuracy and standard deviation of GIB and GIB without $\mathcal{L}_{con}$ on the COLLAB dataset at the 16-dimensional hidden dimension.

SNR		-15 dB	-5 dB	5 dB	15 dB	25 dB
GIB	mean	0.537	0.701	0.760	0.763	0.768
	stdev	0.022	0.012	0.012	0.014	0.013
GIB without $\mathcal{L}_{con}$	mean	0.541	0.699	0.752	0.760	0.763
	stdev	0.018	0.024	0.016	0.016	0.013

TABLE VII: Influence of Mutual Information Weight β on Graph Classification Accuracy

β \ SNR	-15 dB	-5 dB	5 dB	15 dB	25 dB
0.01	0.530	0.648	0.706	0.722	0.731
0.1	0.564	**0.661**	0.720	**0.734**	**0.738**
0.3	**0.575**	0.650	**0.727**	0.730	0.733
0.5	0.552	0.660	0.718	0.724	0.727
0.7	0.571	0.643	0.722	0.728	0.736

1) Effectiveness of $\mathcal{L}_{MI}$: Observations from Tables III and IV indicate that, across both datasets, the system based on GIB and its variant without the $\mathcal{L}_{MI}$ component consistently exhibit superior performance at a hidden dimension of 32 compared to a hidden dimension of 16. This is because an increase in the hidden layer dimension increases the capacity of the model, which means that the network can capture more details in the data. Especially for graph data with complex structures, higher model capacity can help the model learn richer representations and thus improve performance.

It is worth noting that the size of hidden dimension has a more significant effect on the performance of the variant without $\mathcal{L}_{MI}$ than GIB. The absence of $\mathcal{L}_{MI}$ in the variant causes the model to focus more on maximizing mutual information between the input graph data and the extracted representations, while neglecting the aspect of compression. Consequently, the extracted representations contain much redundant information that takes up space for useful information. Therefore, as the hidden dimension increases, the model capacity increases. This allows the model to capture richer information in the graph data and more information needed for the task. In other words, GIB takes into account information compression, meaning it can achieve good performance even with relatively smaller model capacity.

2) Effectiveness of $\mathcal{L}_{con}$: Furthermore, the absence of the connectivity loss constraint, $\mathcal{L}_{con}$, not only detrimentally impacts system performance on certain datasets but also instigates volatility in node assignments, subsequently leading to inconsistent graph classification results. To illustrate the

significance of connectivity loss and its impact on system reliability, we adopt the arithmetic mean of outcomes derived from a 10-fold cross-validation process as the benchmark for task execution accuracy. Complementarily, the standard deviation computed from these same 10 iterations serves as the metric to gauge the consistency and robustness of our system's performance under different communication conditions. It becomes evident upon analysis that the standard deviation associated with GIB added $\mathcal{L}_{con}$ exhibits a reduction in comparison to its variant without $\mathcal{L}_{con}$ under various channel conditions. This result suggests that the connectivity loss we incorporate plays a critical role in improving the robustness and stability of the system performance.

3) Influence of Variations in β on System Performance: The hyperparameter β, associated with the mutual information term, plays a pivotal role in modulating the extent of mutual information between the graph features and the subgraph features, thereby reflecting the relative importance of the graph's global structure versus its local features. To explore this interplay, we conducted experiments assessing the impact of

varying β values on classification accuracy, with the findings presented in Table VII. The experimental data reveal that the system attains optimal performance at a moderate MI factor level, in contrast to either extremely high or low levels.

When β is set to a low value, the model predominantly accentuates local features, while marginalizing the global structural information of the graph. This can lead to the model's failure in capturing critical global relationships within the graph, culminating in a diminished classification accuracy. Inversely, a high β value results in the model prioritizing the global graph structure at the expense of local features. Such an overemphasis on the macro structure of the graph renders the classifier less receptive to nuanced local features, thereby constraining the model's task execution capabilities. By selecting a moderate value for β, the model achieves an equilibrium between harnessing the global structure and incorporating local features. This equilibrium enables the model to effectively utilize the intrinsic global structure of the graph while concurrently considering the node-specific local features. The resultant synergistic integration of global and local features thus significantly enhances the model's performance in graph classification tasks.

VI. DISCUSSION

Task-oriented graph data transmission is a promising research avenue, especially relevant in smart city scenarios such as traffic management and environmental monitoring, where distributed edge computing plays a crucial role. In these contexts, numerous sensors and devices across the city communicate with central servers for data processing. This communication technique significantly reduces bandwidth usage while maintaining accurate inference capabilities, making it highly suitable for real-time applications in urban settings. To effectively translate the theoretical foundations of the GIB approach into tangible applications, it is essential to demonstrate its practical utility in addressing real-world, task-oriented communication challenges. Here are some heuristics worth exploring.

A. Heuristics for Inference Goal Formulation

Real-world scenarios encompass a broad spectrum of inference objectives, contingent upon the specific application context. For example, in the realm of autonomous vehicles, the focus might be on detecting and responding to traffic signals and obstacles. In contrast, industrial IoT may concentrate on predicting equipment failures. The GIB framework is designed to fine-tune the communication process, thereby amplifying the precision and efficiency tailored to these particular tasks. The GIB model can be crafted to cater to either uni-task or multi-tasks. While task-specific models are optimized for a unique objective, task-agnostic models can be expanded to address a variety of tasks through strategic architectural enhancements and transfer learning techniques. This approach necessitates training the model across a diverse array of tasks and datasets, fostering its ability to generalize across varied communication scenarios.

B. Heuristics for GIB Model Training

The GIB model should be trained on data that mirrors the complexities of its target real-world task. This process entails amassing comprehensive datasets that encompass both raw input data, such as sensor readings or visual imagery, and the relevant task-specific labels, such as classifications of objects or indicators of faults. It is imperative that these data are meticulously annotated and preprocessed to construct a graph that encapsulates the interconnections and interdependencies among various data elements. The efficacy of GIB framework should be substantiated through rigorous experimentation with datasets that pertain to different task-oriented communication applications. For instance, industrial datasets could be utilized to train the model in predictive maintenance tasks, employing sensor data to anticipate equipment failures. Additionally, deploying the GIB framework on medical imaging datasets could enhance communication efficiency in tasks such as diagnosing diseases and monitoring patient conditions.

C. Heuristics for Training Dataset Construction

To construct a training dataset for tasks without a standard dataset, follow these steps: First, clearly specify the task, such as node classification or link prediction; second, collect raw data needed for the graph-based task, typically in the form of nodes, edges, and node features; third, preprocess the data suitable for graph-based learning by normalizing node features, removing self-loops, and converting the graph to an appropriate format (e.g., adjacency matrix, edge list); Construct graph representations to serve as inputs to GNN, typically involving adjacency matrices and node feature matrices; Generate pairs of input graphs and target outputs specific to the task. For instance, for node classification, the input is subgraphs centered around each node with the output being the class label of the central node for node classification tasks, while for link prediction tasks, the input is subgraphs around a candidate link with the target being a binary label indicating whether the link exists; Implement GIB to identify and retain the most informative parts of the graph while discarding unnecessary information by encoding the graph data into a compressed representation and decoding it to approximate the original or task-specific output; Finally, divide the dataset to ensure the model generalizes well to unseen data. Additionally, feature engineering, such as aggregating neighborhood information and generating higher-order features, can enrich node features. Data augmentation can also be used to improve the model's robustness in data-hungry cases.

VII. CONCLUSION

In this study, we explored GIB-based task-oriented communication for graph data transmission, focusing on optimizing mutual information between received codewords and the task goal while reducing the mutual information with the original graph representation. Utilizing variational approximation, Monte Carlo sampling, and MINE, we devised a workable objective function, addressing the challenge of computing mutual information for irregular graph data. We also incorporated a connectivity loss term to account for community structure in

graph data, which improved subgraph selection and stabilized training, as confirmed by our experiments. Furthermore, we adapted our method for digital transmission using vector quantization. Our tests across various SDC channel qualities consistently showed strong performance, underscoring the method's adaptability and efficacy. Future work will explore more practical application areas and include hands-on experiments with real-world data.

REFERENCES

[1] M. Mohsenivatani, S. Ali, V. Ranasinghe, N. Rajatheva, and M. Latva-Aho, "Graph representation learning for wireless communications," *IEEE Commun. Mag.*, vol. 62, no. 1, pp. 141–147, 2024.

[2] Y. Tian, Y. Niu, J. Yan, and F. Tian, "Inferring private attributes based on graph convolutional neural network in social networks," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Daegu, South Korea, Mar. 2019, pp. 186–190.

[3] S. Shaikh, S. Rathi, and P. Janrao, "Recommendation system in e-commerce websites: A graph based approached," in *Proc. IEEE 7th Int. Advance Comput. Conf. (IACC)*, Daegu, Korea, Jul. 2017, pp. 931–934.

[4] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Sep. 2017.

[5] Y. Duan, L. Shao, G. Hu, Z. Zhou, Q. Zou, and Z. Lin, "Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph," in *Proc. IEEE 15th Int. Conf. Softw. Eng. Res. Manage. Appl. (SERA)*, London, United Kingdom, Jul. 2017, pp. 327–332.

[6] W. W. M. Lam and K. C. C. Chan, "A graph mining algorithm for classifying chemical compounds," in *Proc. IEEE Int. Conf. Bioinf. Biomedicine*, Philadephia, Pennsylvania, USA, Nov. 2008, pp. 321–324.

[7] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *ArXiv: 1506.05163*, Jun. 2015.

[8] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond transmitting bits: Context, semantics, and task-oriented communications," *IEEE J. Sel. Areas Commun.*, vol. 41, pp. 5–41, Nov. 2022.

[9] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, Apr. 2021.

[10] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, pp. 210–219, Jan. 2022.

[11] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE J. Sel. Areas Commun.*, vol. 40, pp. 197–211, Nov. 2021.

[12] Z. Weng, Z. Qin, and G. Y. Li, "Semantic communications for speech recognition," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Feb. 2022, pp. 1–6.

[13] J. Shao, Y. Mao, and J. Zhang, "Task-oriented communication for multidevice cooperative edge inference," *IEEE Trans. Wireless Commun.*, vol. 22, pp. 73–87, Jul. 2022.

[14] N. Farsad, M. Rao, and A. J. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Sep. 2018, pp. 2326–2330.

[15] S. Guo, Y. Wang, S. Li, and N. Saeed, "Semantic importance-aware communications using pre-trained language models," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2328–2332, Jul. 2023.

[16] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, pp. 567–579, Apr. 2018.

[20] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 109–127, Apr. 2021.

[17] D. B. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, pp. 178–193, Apr. 2020.

[18] X. Kang, B. Song, J. Guo, Z. Qin, and F. R. Yu, "Task-oriented image transmission for scene classification in unmanned aerial systems," *IEEE Trans. Commun.*, vol. 70, pp. 5181–5192, Jun. 2022.

[19] P. Li, N. Shlezinger, H. Zhang, B. Wang, and Y. C. Eldar, "Graph signal compression by joint quantization and sampling," *IEEE Trans. Signal Process.*, vol. 70, pp. 4512–4527, Sep. 2022.

[21] Y. Wang, S. Guo, Y. Deng, H. Zhang, and Y. Fang, "Privacy-preserving task-oriented semantic communications against model inversion attacks," *IEEE Trans. Wireless Commun.*, pp. 1–1, Mar. 2024.

[22] N. Tishby, F. C. N. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, Oct. 1999, p. 368–377.

[23] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jerusalem, Israel, Jun. 2015, pp. 1–5.

[24] Q. Wu, H. Zhang, J. Yan, and D. P. Wipf, "Handling distribution shifts on graphs: An invariance perspective," in *Proc. Int. Conf. Learn. Represent., ICLR*, Virtual, Feb. 2022.

[25] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. S. Quek, and G. Min, "Mobility-aware cluster federated learning in hierarchical wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8441–8458, Oct. 2022.

[26] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in *Advances Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Virtual, 2020, pp. 437–448.

[27] 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, pp. 4–24, Mar. 2020.

[28] G. Zhao, Q. Wang, F. Yao, Y. Zhang, and G. Yu, "Survey on large-scale graph neural network systems," *J. Software*, vol. 33, no. 1, pp. 150–170, 2022.

[29] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Represent., ICLR*, Apr. 2017, pp. 1–19.

[30] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, R. D. Hjelm, and A. C. Courville, "Mutual information neural estimation," in *Proc. Int. Conf. on Mach. Learn. (ICML)*, vol. 80, Stockholmsmässan, Stockholm, Sweden, Jul. 2018, pp. 530–539.

[31] J. Yu, T. Xu, Y. Rong, Y. Bian, J. Huang, and R. He, "Graph information bottleneck for subgraph recognition," in *Proc. Int. Conf. Learn. Represent., ICLR*, Oct. 2020.

[32] S. Xie, S. Ma, M. Ding, Y. Shi, M.-F. Tang, and Y. Wu, "Robust information bottleneck for task-oriented communication with digital modulation," *IEEE J. Sel. Areas Commun.*, vol. 41, pp. 2577–2591, Jun. 2023.

[33] Y. Bo, Y. Duan, S. Shao, and M. Tao, "Learning based joint coding-modulation for digital semantic communication systems," *Proc. 14th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, pp. 1–6, Feb. 2023.

[34] Q. Hu, G. Zhang, Z. Qin, Y. Cai, G. Yu, and G. Y. Li, "Robust semantic communications with masked VQ-VAE enabled codebook," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8707–8722, Apr. 2023.

[35] F. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Represent., ICLR*, Addis Ababa, Ethiopia, Apr. 2020.

[36] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int. Conf. Learn. Represent., ICLR*, New Orleans, LA, USA, May 2019.

[37] E. Ranjan, S. Sanyal, and P. P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, New York, NY, USA, Feb. 2019, pp. 5470–5477.