GeNet: A Graph Neural Network-based Anti-noise Task-Oriented Semantic Communication Paradigm

Chunhang Zheng, Kechao Cai
School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, China
zhengchh6@mail2.sysu.edu.cn, caikch3@mail.sysu.edu.cn

***Abstract*—Traditional approaches to semantic communication tasks rely on the knowledge of the signal-to-noise ratio (SNR) to mitigate channel noise. However, these methods necessitate training under specific SNR conditions, entailing considerable time and computational resources. In this paper, we propose GeNet, a Graph Neural Network (GNN)-based paradigm for semantic communication aimed at combating noise, thereby facilitating Task-Oriented Communication (TOC). We propose a novel approach where we first transform the input data image into graph structures. Then we leverage a GNN-based encoder to extract semantic information from the source data. This extracted semantic information is then transmitted through the channel. At the receiver's end, a GNN-based decoder is utilized to reconstruct the relevant semantic information from the source data for TOC. Through experimental evaluation, we show GeNet's effectiveness in anti-noise TOC while decoupling the SNR dependency. We further evaluate GeNet's performance by varying the number of nodes, revealing its versatility as a new paradigm for semantic communication. Additionally, we show GeNet's robustness to geometric transformations by testing it with different rotation angles, without resorting to data augmentation.**

***Index Terms*—Semantic communication, anti-noise, task-oriented communication, graph neural network**

I. Introduction

In an era where traditional communication technologies approach the Shannon limit, the emergence of Artificial Intelligence (AI) has fostered innovative paradigms such as Semantic Communication (SC). Unlike conventional methods that solely focus on accurately transmitting bits, SC delves into the core of transmitted information, aiming to preserve its meaning across the communication channel [1]. This approach is particularly crucial in environments with high network complexity and limited spectrum resources, as it significantly reduces data transmission load, thereby enhancing communication efficiency. Among the various aspects of SC, Task-Oriented Communication (TOC) distinguishes itself by emphasizing the extraction of receiver-relevant semantic information from the source [2], thereby enabling more informed and effective task inference at the receiver's end.

SC has been widely studied in the literature, with most previous methods relying on neural networks (NN) to extract semantic information from the source data and reconstruct it at the receiver's end. As a aspect of SC, TOC provides a unique perspective by focusing on the ultimate goal or task for which the communication is intended, rather than merely ensuring the fidelity of information transmission. This shift towards task-oriented communication encourages the development of systems that are not only efficient in terms of bandwidth usage but also in terms of the computational resources required for processing the transmitted data.

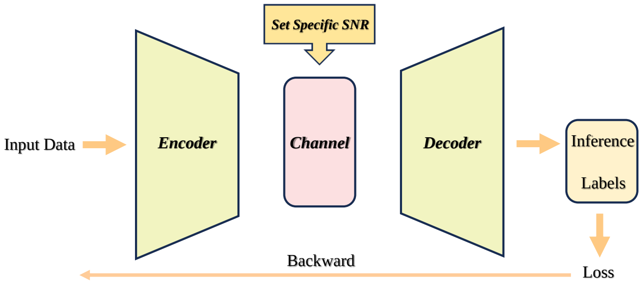


Fig. 1: General semantic communication paradigm. Previous methods need to *set specific SNR*.

However, all previous TOC methods for noise handling in channels rely on knowledge of the SNR. Specifically, these models must be trained under specific SNR conditions, shown in Fig. 1, and their performance may degrade significantly when operating under different SNR conditions than those encountered during training. In this case, models should be trained in different channel conditions to get a satisfied performance, which requires significant time and computational resources.

Moreover, Graph Neural Network (GNN) has emerged as a powerful tool for processing graph-structured data, such as social networks, citation networks, and molecular structures. GNN has been widely applied in various fields, including recommendation systems [3], drug discovery [4], and computer vision. The success of GNN in these fields has inspired researchers to explore its potential in wireless communications. In particular, GNN has been applied to wireless communication problems such as task offloading [5] and cellular traffic prediction [6].

In this paper, we explore a new paradigm for SC, named GeNet, which leverages GNN to facilitate TOC. GeNet is designed to address the challenges of anti-noise TOC, where the transmitted data in a channel is corrupted by noise while decoupling the SNR. Specifically, GeNet comprises a GNN-based encoder and a GNN-based decoder. The GNN-based encoder is designed to extract semantic information from the source data, while the GNN-based decoder is designed to reconstruct the task-relevant semantic information of source data for TOC in downstream. We demonstrate the effectiveness of GeNet in anti-noise TOC through extensive experiments on MNIST [7], FashionMNIST [8] and CIFAR10 [9] datasets. Additionally, our source code has been released and publicly available[1].

In summary, our work makes the following contributions:

- To the best of our knowledge, we are the first to apply GNN to anti-noise semantic communication. We propose a novel GNN-based framework for anti-noise semantic communication, which consists of a GNN-based encoder and a GNN-based decoder, and explored the possibility of it serving as a semantic communication paradigm.
- Our model can denoise on the AWGN channel without knowing the SNR. Specifically, unlike other anti-noise semantic communication methods, our method is able to anti-noise semantic communication while decoupling the SNR. We demonstrate the effectiveness of GeNet in anti-noise TOC through extensive experiments on MNIST, FashionMNIST and CIFAR10 datasets.
- Compared with traditional CNN-based methods, our GNN-based model can process images of different pixels without resizing the image size that may cause information loss or redesigning and retraining the model architecture. This is crucial in scenarios where the image size is not fixed or the image size is large.
- Additionally, our model can extract features that are inherently invariant to geometric transformations such as rotations ensuring that the essence of the transmitted information is preserved and accurately interpreted at the receiving end.

The rest of this paper is organized as follows. Section II introduces the related work on DL-based semantic communication and graph neural network in wireless communications. Section III presents the preprocessing of image data and the proposed GeNet framework. In Section IV, we provide extensive simulation results to evaluate the performance of the proposed methods. Finally, Section V concludes the paper.

II. Related Work

A. DL-Based Semantic Communication

The purpose of the next generation of communication systems has evolved from reliable transmission of bit sequences to a wider range of objectives, namely providing intelligent, efficient, and sustainable communication services. Its core is to extract the message semantics of the transmitting end based on the task requirements of the receiving end, and to achieve more intelligent communication and interaction based on the context and requirements of the receiving end, aiming to interpret the meaning of the transmitted information accurately across the communication channel to improve the efficiency of communication systems.

DL techniques have shown great potential in processing various intelligent tasks, i.e., computer vision and NLP [10]. In the communication area, the recent line of research on semantic communication [11]–[14] has motivated a paradigm shift in communication system design for many types of data.

In particular, a task-oriented semantic communication system was proposed in [11] that provides text and speech of the target language at the receiving end by transmitting low dimensional semantic features to achieve translation tasks in speech transmission, and exhibits higher robustness in the case of channel damage. A task-oriented communication for multi-device cooperative edge inference was proposed in [14], where a group of low-end edge devices transmit the task-relevant features to an edge server for aggregation and processing. In the seminal work of semantic communication [12], E. Bourtsoulatze et al. construct an encoder-decoder architecture through neural networks and uses the energy constrained encoder outputs. The output of the beam encoder is transmitted through AWGN and Rayleigh fading channels, achieving comparative transmission in noisy environments Better performance of unified image encoding JPEG and JPEG2000.

B. Graph Neural Network in Wireless Communication

GNN have emerged as a focal point in research due to their remarkable learning capability on graph-structured data across diverse applications. As an extension of neural networks, GNN can handle data formats represented by graph structures. In the graph, each node is defined by its own features and its adjacent nodes and relationships, and the network calculates the node's representation vector by recursively aggregating and transforming the representation vectors of adjacent nodes. The goal of GNN is to learn useful representations from graph structured data and use these representations for various tasks, such as node classification, graph classification, link prediction, and so on.

GNN is originated from seminal work by [15], which introduced the Graph Fourier Transform, extending the Fourier transform to graph structures and paving the way for spectral graph convolution networks. Foundational research in GNN includes the introduction of spectral-based Graph Convolutional Network (GCN) by [16], leveraging the Graph Fourier Transform. Furthermore, attention mechanisms, as seen in [17], autonomously learn the importance of edge relationships between nodes. In applied GNN research, [18] proposed a method for dynamic WLAN performance prediction using a heterogeneous temporal graph neural network, while [19] employed GNN and GLUs to extract spatial and temporal information, respectively, for traffic prediction tasks in cellular networks.

III. GNN-Based SC Framework

Most models for semantic communication, specifically task-oriented communication, have an encoder-decoder structure [12]–[14]. Here, the input data representations $(x_1, ..., x_n)$ are mapped to a low-dimensional semantic space $(z_1, ..., z_m)$ by the encoder, and the decoder maps the low-dimensional semantic information to the data representations $(\hat{y}_1, ..., \hat{y}_o)$ which is relevant to the receiver for downstream task. The

[1] https://github.com/chunbaobao/GeNet

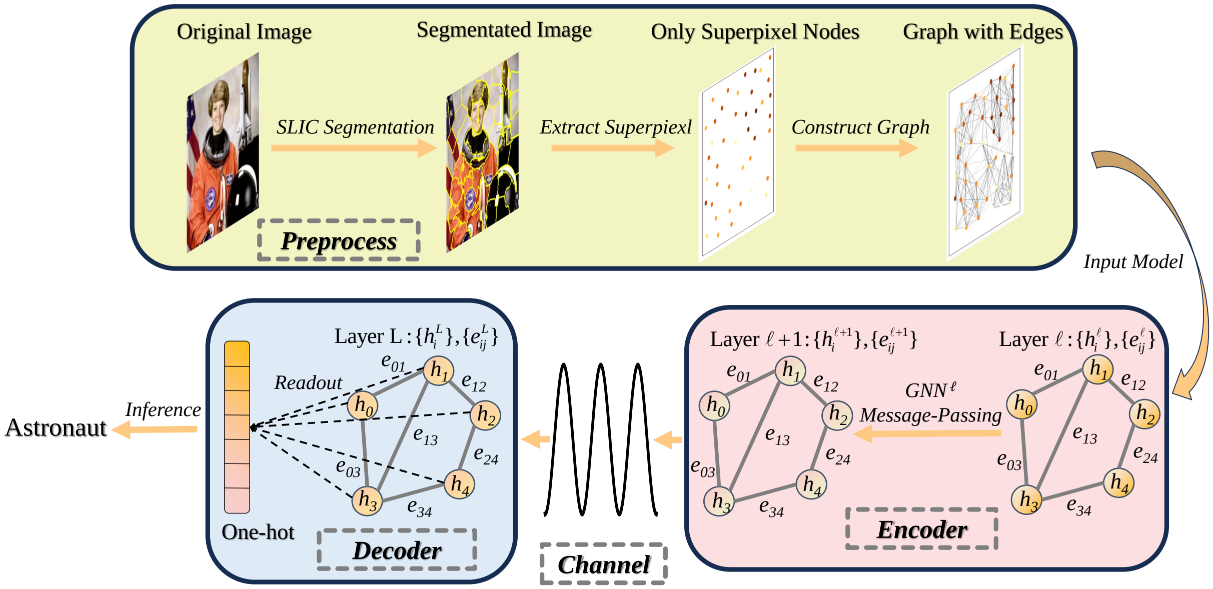


Fig. 2: Our GeNet model architecture.

encoder and decoder are trained to minimize the error between the inference and task-relevant labels $(y_1, ..., y_o)$.

Our proposed GeNet framework follows this overall architecture that consists of a GNN-based encoder and a GNN-based decoder, shown in the Fig. 2. The details of the GeNet model are described in the following sections.

A. Preprocessing

The preprocessing comprises two primary steps. First, we extract the segmentation of the input image. Second, we transform the segmentation information into the graph structure.

In the first step, we use SLIC segmentation method [20]–[22] to segment the popular MNIST, FashionMNIST and CIFAR10 image classification datasets. SLIC is a simple and efficient method for image segmentation, which is based on the K-means clustering algorithm. It is able to generate superpixels that are more regular in shape and size, and it is also able to preserve the boundaries of the objects in the image.

In the second step, we transform the segmentation into a graph structure similar to [23]. Specifically, we use the superpixels as the nodes of the graph, and the features of the nodes are the average color and the centroid of the superpixels. The edges are connected between the nodes and their k-th nearest neighbors, and the weight of the edge is determined by the similarity of the nodes which is calculated by Euclidean distance between the the features of the nodes:

$$A_{ij}^k = \exp\left(-\frac{\|\mathcal{N}_i - \mathcal{N}_j\|^2}{\sigma_i^2}\right), \tag{1}$$

$$\sigma_i = \sum_1^k \|\mathcal{D}_i^k\|^2 + \epsilon, \tag{2}$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ are the features of the superpixel nodes, ϵ is a small constant to avoid division by zero, and σ_i is the scale parameter. And the $\mathcal{D}_i^k$ is the similarity between the node i and its k-th nearest neighbor.

The superpixel graph generation procedures from image are summarized in Algorithm 1. In this algorithm, we first compute the features of each nodes that are the average color and the centroid of the superpixels. We then construct the adjacency matrix based on Euclidean distance between the the features of the nodes.

Algorithm 1 Superpixel Graph Generation from Segmentation

```
Input: Origial Image $I$, segmentation mask $S$, number of nodes $N$.
Output: Superpixel Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
Initialization: Set $\mathcal{N} \leftarrow \emptyset$, $\mathcal{E} \leftarrow \emptyset$
for $n = 1$ to $N$ do
    Compute color feature $c_n$ as the average color of superpixel $S(n)$ in $I$.
    Compute coordinate feature $p_n$ as the centroid of superpixel $S(n)$ in $I$.
    Add node $n$ to $\mathcal{N}$ with features $\mathcal{N}(n) = [c_n, p_n]$
end for
Computer adjacency matrix $\mathcal{E}$ based on (1) and (2).
Output: Superpixel graph $\mathcal{G}$ with nodes $\mathcal{N}$ and edges $\mathcal{E}$.
```

B. Encoder and Decoder

There are two representive GNN frameworks: Message Passing Neural Network (MPNN) [24] and Weisfeiler Lehman

Network GNNs (WL-GNNs). MPNNs operate by passing messages between nodes in a graph, allowing nodes to aggregate and update information from neighboring nodes iteratively. They are versatile and widely used in tasks such as graph classification, node classification, and link prediction. WL-GNNs, on the other hand, are based on the Weisfeiler-Lehman (WL) graph isomorphism test, which is a powerful tool for distinguishing non-isomorphic graphs.

In this work, we use the MPNN architecture as the backbone of our GeNet model. The MPNN architecture consists of two main components: message passing and readout. The message passing component is used to update the node features based on the features of the neighboring nodes:

$$\mathcal{M}_e^{(\ell+1)} = \phi\left(\mathcal{N}_v^{(\ell)}, \mathcal{N}_u^{(\ell)}, \mathcal{E}_e^{(\ell)}\right), (u, v, e) \in \mathcal{G}, \tag{3}$$

$$\mathcal{N}_v^{(\ell+1)} = \psi\left(\mathcal{N}_v^{(\ell)}, \rho\left(\left\{\mathcal{M}_e^{(\ell+1)} : (u, v, e) \in \mathcal{G}\right\}\right)\right), \tag{4}$$

where $\mathcal{N}_v^{(\ell)}$ and $\mathcal{N}_u^{(\ell)}$ are the features of the nodes v and u at the ℓ-th layers, respectively, and $\mathcal{E}_e^{(\ell)}$ is the feature of the edge e. We denote ϕ as the message function defined for every edge in the graph. ρ denotes the aggregation function that aggregates the messages received by nodes; ψ denotes the update function that updates the node features based on the aggregated messages and the node features itself.

The readout component is used to aggregate the node features in the last layer to obtain the graph-level features:

$$h_{\mathcal{G}} = \rho_{\mathcal{G}}\left(\left\{\mathcal{N}_v^{(L)} : v \in \mathcal{N}\right\}\right), \tag{5}$$

where $h_{\mathcal{G}}$ is the graph-level features, L is the number if stacked GNN layers, and $\rho_{\mathcal{G}}$ is the readout function to obtain the graph-level features. The graph-level features are then used to make the final prediction.

In our GeNet model, the encoder and decoder correspond to the message passing and readout components of the MPNN architecture, respectively. The encoder is used to extract the semantic information from the source data based on (3) and (4), and the decoder is used to reconstruct the interested semantic information of source data for TOC based on (5).

As for the channel, we use the AWGN channel to simulate the noise in the communication. We add noise to node features using Algorithm 2. In this algorithm, we first compute the channel dimension and the signal power of the graph, and then we compute the noise power based on the SNR, which is defined as follows:

$$\mathrm{SNR} = 10 \log_{10} \frac{P}{\sigma^2} \quad (\mathrm{dB}). \tag{6}$$

We then sample the noise from the Gaussian distribution and add the noise to the node features. The noisy node features are then used as the input to the decoder. It is worth noting that since the decoder directly performs readout processing on node features to obtain graph-level features, there is no need to simulate noise addition to the edges of the graph in the channel.

Algorithm 2 Add Noise to Node Features

```
**Input:** Node features $\mathcal{N} \in \mathbb{R}^{N \times d}$, noise level SNR (dB).
**Output:** Noisy Node features $\mathcal{N}'$
Compute the channel dimension $k = N \times d$
Compute the signal power $P = \frac{\sum_1^N \sum_1^d \mathcal{N}_{ij}^2}{k}$
Compute noise power $\mathrm{SNR} = 10 \log_{10} \frac{P}{\sigma_n^2}$ based on (6)
**For** $v = 1$ to $N$ **do**
    Sample noise $n_v \sim \mathcal{N}(0, \sigma_n^2)$ with $n_v \in \mathbb{R}^d$
    Add noise to node $v$ $\mathcal{N}_v' = \mathcal{N}_v + n_v$
**end for**
**Output:** Noisy Node features $\mathcal{N}'$
```

In our GeNet model, we use the mean readout function to aggregate the node features in the decoder to obtain the graph-level features, which can reduce the noise in the channel. The mean readout function is defined as:

$$h_{\mathcal{G}} = \frac{1}{N} \sum_{v \in \mathcal{N}} \mathcal{N}_v^{(L)}. \tag{7}$$

Assuming that the noise is independent and identically distributed (i.i.d.). As mentioned in Algorithm 2, the noisy node feature i is $\mathcal{N}_i' = \mathcal{N}_i + n_i$. Given that the mean readout function, we have the following graph-level features when the noise is added to the node features:

$$\begin{aligned} h_{\mathcal{G}} &= \frac{1}{N} \sum_{v \in \mathcal{N}} \left(\mathcal{N}_v^{(L)} + n_v\right) \\ &= \frac{1}{N} \sum_{v \in \mathcal{N}} \mathcal{N}_v^{(L)} + \frac{1}{N} \sum_{v \in \mathcal{N}} n_v. \end{aligned} \tag{8}$$

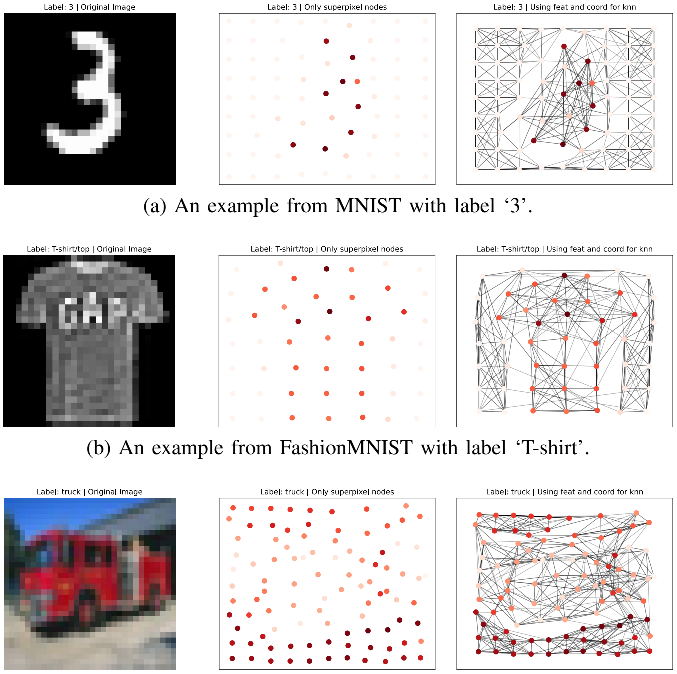
Among them, $n_v \sim \mathcal{N}(0, \sigma_n^2)$, and $\frac{1}{N} \sum_{v \in \mathcal{N}} n_v \sim \mathcal{N}(0, \frac{\sigma_n^2}{N})$. Therefore, the noise power in the graph-level features is reduced by a factor of $\frac{1}{N}$ compared to the noise power in the channel.

IV. Experiments

In this section, we assess the performance of the proposed GeNet model on the MNIST [7], FashionMNIST [8] and CIFAR10 [9] datasets. And all experiments are conducted either on a computer equipped with a single NVIDIA GeForce RTX 3090 GPU and an Intel Core i7-12700K or one with two NVIDIA GeForce RTX 3090 GPUs and an Intel Xeon Gold 6230. For the development, we use the PyTorch [25] and Deep Graph Library (DGL) [26].

A. Experiment Setup

1) Preprocessing: The MNIST dataset comprises 60,000 training images and 10,000 test images of handwritten digits for 10 classes, ranging from 0 to 9. Each image is grayscale and has a resolution of 28x28 pixels. The FashionMNIST dataset consisting of 60,000 training images and 10,000 test


Label: 3 | Original Image
Label: 3 | Only superpixel nodes
Label: 3 | Using feat and coord for knn
Label: T-shirt/top | Original Image
Label: T-shirt/top | Only superpixel nodes
Label: T-shirt/top | Using feat and coord for knn
Label: truck | Original Image
Label: truck | Only superpixel nodes
Label: truck | Using feat and coord for knn

(a) An example from MNIST with label ‘3’.

(b) An example from FashionMNIST with label ‘T-shirt’.

(c) An example from CIFAR10 with label ‘truck’.

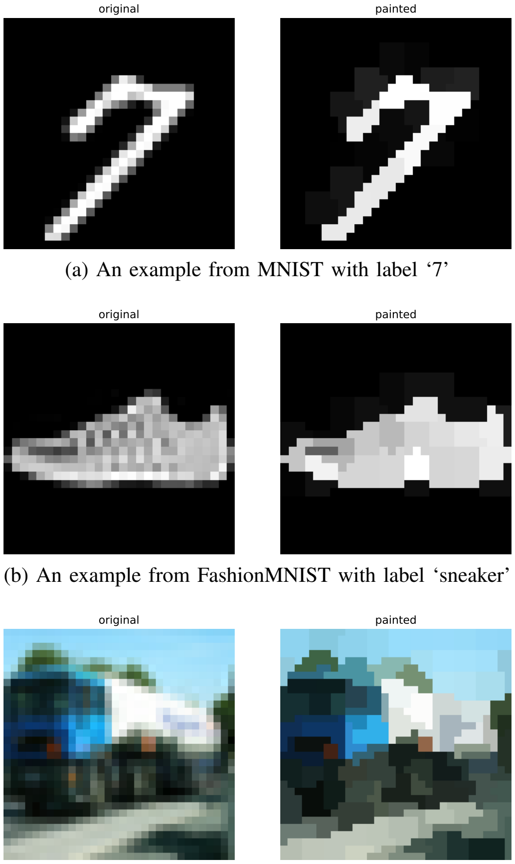
Fig. 3: Examples of the superpixel graph generation

images of 10 classes of fashion items, shares the same resolution as MNIST. On the other hand, the CIFAR10 dataset containing 50,000 training images and 10,000 test images of 10 classes, features RGB images with a resolution of 32x32 pixels.

To create a validation set, 10% of each class from the training dataset is randomly selected for all datasets. It should be noted that the ‘6’ and ‘9’ classes in MNIST are excluded from the validation and test datasets as their rotated versions are identical. Following the approach in previous works [22], [23], specific parameter settings are applied: 95 nodes and a compactness parameter of 0.25 for MNIST, 105 nodes and a compactness parameter of 0.3 for FashionMNIST, and 200 nodes with a compactness parameter of 10 for CIFAR10[2].

Furthermore, exemplifying the superpixel graph generation process from the image in Fig. 3, the left sub-image represents the original image while the middle sub-image illustrates the superpixel nodes generated by SLIC, incorporating average color and centroid position information. The right sub-image showcases the superpixel graph with its adjacency matrix generated using Algorithm 1.

2) Training: To fully demonstrate the effectiveness of our proposed model, we used various GNN models as the backbone of the GeNet model, including Graph Convolutional Network (GCN) [16], Graph Attention Network (GAT) [17], Gated Graph Convolutional Network (GATEDGCN) [27], along with GNNs augmented with Multi-layer Perceptrons (MLP). MLP does not directly use the original image as input, but uses node features converted to a graph structure as input.



(a) An example from MNIST with label ‘7’

(b) An example from FashionMNIST with label ‘sneaker’

(c) An example from CIFAR10 with label ‘truck’

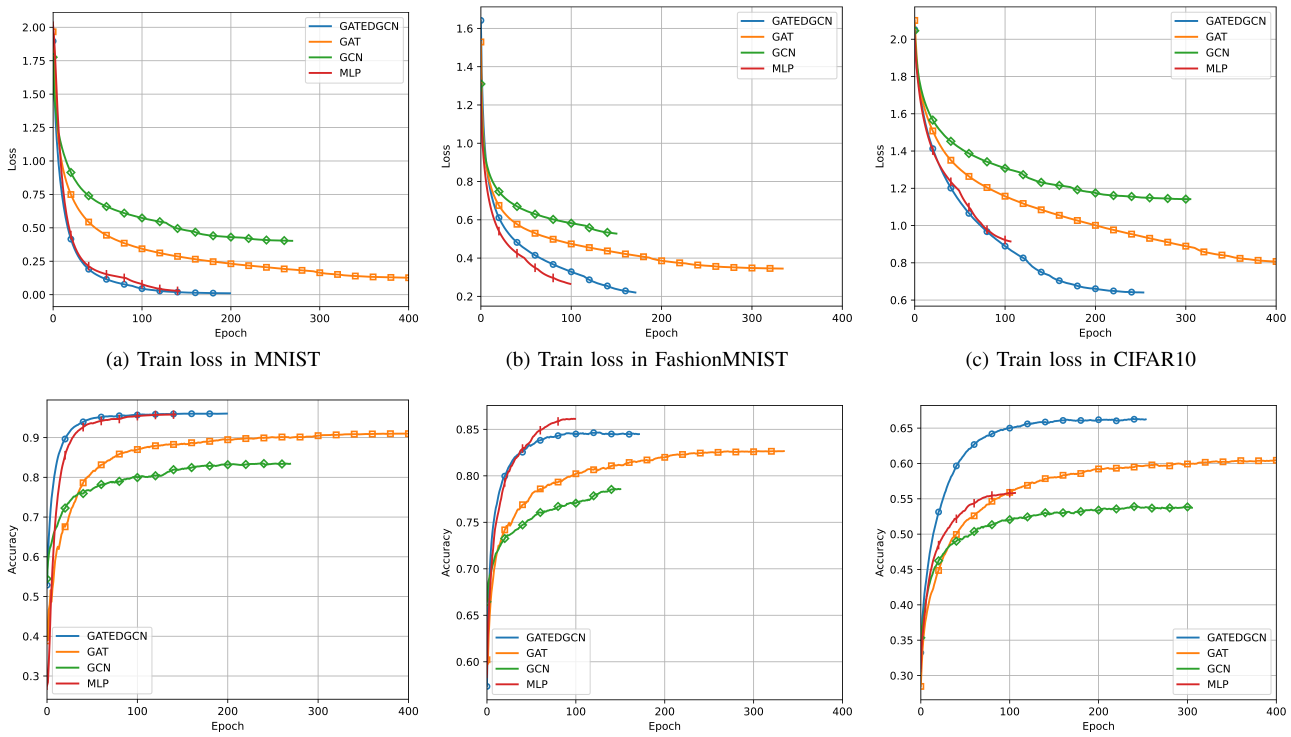
Fig. 4: Examples showing the loss of color information.

The models were all configured with a consistent architecture comprising 4 stacked GNN layers. Training was conducted using the Adam optimizer [28] with an initial learning rate, which was dynamically adjusted by reducing it when the validation set accuracy failed to improve over consecutive epochs. Training ceased once the learning rate reached a specified threshold to avoid overfitting. To ensure that the model was not underfit, an ample number of training epochs were established. The cross-entropy loss function was employed to quantify the disparity between the predicted outcomes and the relevant task labels. Further elaboration on the model structures and training setups is available in Table I.

As for the baseline, there are only few works that are relevant to GNN-based semantic communication. We compare our proposed GeNet model with baselines as follows:

- ResNet [29]: This baseline is trained and tested on the original image dataset, serves as a benchmark representing the upper bound of GeNet’s potential performance.
- ResNet with paint [21]: While GNN models only have access to the average colour and the centroid position, not knowing anything about the superpixel’s shape. We use the paint method to simulate the color information loss of the superpixel, shown in Fig. 4. Specifically, we paint the

[2] https://github.com/bknyaz/graph_attention_pool



(a) Train loss in MNIST (b) Train loss in FashionMNIST (c) Train loss in CIFAR10

(d) Validation accuracy in MNIST (e) Train loss in FashionMNIST (f) Validation accuracy in CIFAR10

Fig. 5: Training loss and validation accuracy of GeNet with different GNN models.

TABLE I: Parameters of models and training configurations.

	GatedGCN	GCN	GAT	MLP
Batch Size	5	5	50	5
Initial Learning Rate	5e-5	5e-5	5e-5	5e-4
LR Reduce Factor	0.5	0.5	0.5	0.5
LR Schedule Patience	25	25	25	10
Min LR	1e-6	1e-6	1e-6	1e-5
Number of Layer	4	4	4	4
Hidden Dim	70	146	19	150
Output Dim	70	146	152	150
Readout	mean	mean	mean	mean
Self Loop	No	No	No	No
Number of Heads	-	-	8	-

original image with the average color of the superpixel. This baseline also is trained on the original image dataset but tested on the painted images.

Both ResNet20 and GNN models possess a comparable number of parameters, which is summarized in Table II. In practice, the performance of ResNet20 was suboptimal on the FashionMNIST dataset. Therefore, a pre-trained ResNet18 model was utilized instead.

B. Results

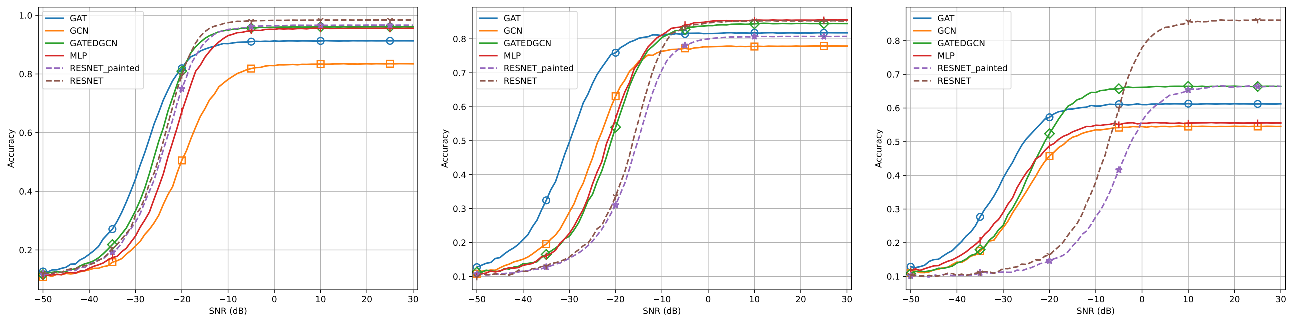
1) Train: To assess the performance disparities of GNN models, we initially examine the GeNet model's efficacy across the MNIST, FashionMNIST, and CIFAR10 datasets with different GNN models as backbones. Fig. 5 presents the train loss and validation accuracy of the GeNet model with the configurations in Table I. It's worth noting that the number of epochs trained for each model may vary slightly. This is due to the reduction of the learning rate when the validation accuracy fails to improve over several epochs, and the training process halts once the learning rate reaches a certain threshold in Section IV-A2.

TABLE II: Number of parameters used in different models.

	GatedGCN	GCN	GAT	MLP	ResNet
MNIST	104,217	101,365	111,008	105,717	41,610
FashionMNIST	104,217	101,365	111,008	105,717	1,117,281
CIFAR10	104,357	101,657	111,312	106,017	269,722

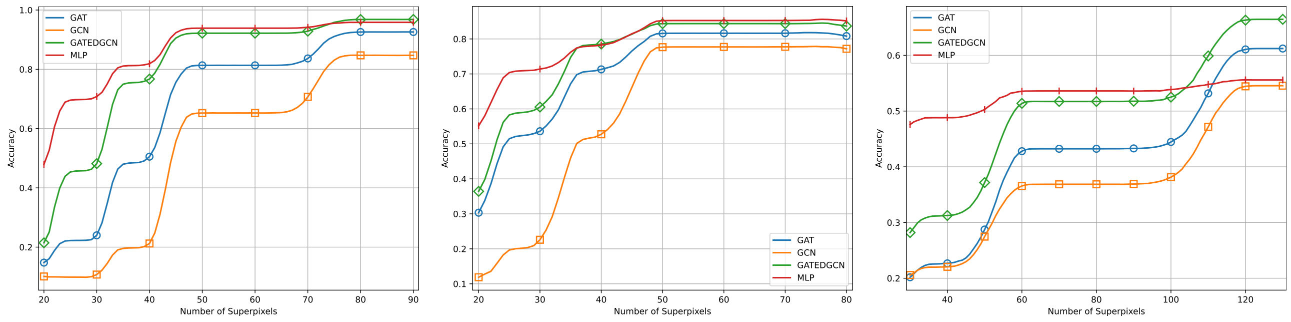
The convergence epochs and accuracy levels demonstrate differences across the GeNet model's various backbones, owing to their distinct expressive capabilities and computational complexities. Notably, the GatedGCN model performs well on all datasets, while the MLP backbone demonstrates rapid convergence. In contrast, the GCN model exhibits subpar performance on both datasets, potentially attributed to inadequate information exchange among graph nodes.

2) Evaluation with SNR: We subsequently assessed the performance of the GeNet model on the test sets of MNIST, FashionMNIST and CIFAR10 datasets across varying SNR



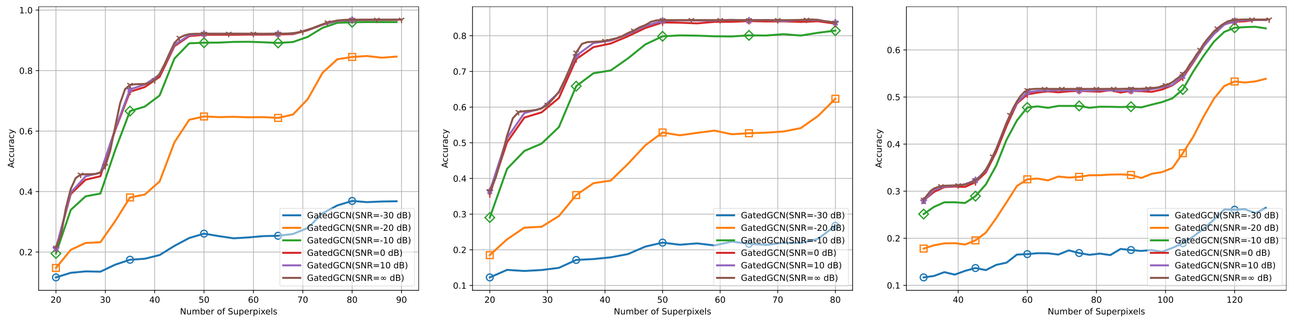
(a) Test accuracy in MNIST (b) Test accuracy in FashionMNIST (c) Test accuracy in CIFAR10

Fig. 6: Test accuracy versus SNR of different models on three test datasets.



(a) Test accuracy in MNIST (b) Test accuracy in FashionMNIST (c) Test accuracy in CIFAR10

Fig. 7: Test accuracy versus number of superpixels of different models on three test datasets.



(a) Test accuracy in MNIST (b) Test accuracy in FashionMNIST (c) Test accuracy in CIFAR10

Fig. 8: Test accuracy versus number of superpixels of GatedGCN under different SNRs on three test datasets.

levels, aiming to ascertain its effectiveness in mitigating noise in task-oriented communication. The outcomes of this evaluation are illustrated in Fig. 6.

Notably, our model exhibits superior performance in low SNR environments compared to baseline models. However, as the SNR increases, compared with the ResNet model as the upper bound, the performance of the GeNet model is still not satisfactory especially in CIFAR10 dataset. This may be due to the loss of information in the superpixel graph structure. This disparity in performance could potentially be attributed to the information loss inherent in the superpixel graph structure. The GeNet shows marginal improvement in noise resistance compared to the baseline on the MNIST dataset. This outcome could be attributed to the MNIST's inherent simplicity, where the influence of noise is relatively minimal.

3) Evaluation with number of superpixel nodes: We further assess the performance of the GeNet model on the test sets of MNIST, FashionMNIST and CIFAR10 datasets, varying the number of superpixel nodes to explore the possibility of it serving as a semantic communication paradigm. The results are shown in Fig. 7. The model's performance demonstrates enhancement with an increase in the expected number of superpixel nodes.

Moreover, we also conducted the evaluation with the num-

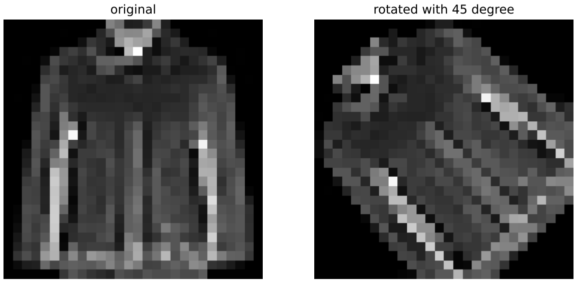


Fig. 9: An example of rotated image from FashionMNIST.

ber of superpixel nodes for different SNR levels on the test sets. The results are shown in Fig. 8. We notice a trend where the reduction in the number of nodes correlates with a decline in the test set accuracy. This phenomenon can be attributed to two primary factors: firstly, a lower number of nodes results in diminished extraction of semantic information; secondly, according to the equation (8), the SNR is inversely proportional to the number of nodes, and the noise level increases as the number of nodes decreases. This trend is consistent across all datasets and all GNN models. Based on this observation, we can do a trade-off in real-world communication systems by adjusting the number of superpixel nodes to balance the communication overhead and task accuracy.

Furthermore, Fig. 8 also illustrates that the performance of the model at an SNR of 0 dB is nearly identical to its performance in the absence of noise. Similarly, at an SNR of -10 dB, the model's effectiveness remains closely matched to the noise-free scenario. This indicates indirectly that the GeNet model has a certain ability to resist noise, and the performance of the model is not significantly affected by noise when the SNR is relatively low.

4) Evaluation with rotation angles: Moreover, as an incidental outcome of this study, we also explored task-oriented communication systems for images with geometric transformations. This facet allows the communication system to accurately infer tasks even when confronted with image data subjected to geometric alterations like rotation or translation which is shown in Fig. 9. To substantiate this capability, we conducted evaluations of the GeNet model on test sets of MNIST, FashionMNIST and CIFAR10 datasets, varying the rotation angles. The results are similar for all datasets, only the CIFAR10 dataset is shown in Fig. 10.

When the rotation angle aligns precisely with an integer multiple of 90 degrees, a noticeable improvement in accuracy on the test set is observed. This enhancement can be linked to the fact that, under these particular conditions, the images post-rotation maintain a greater level of regularity and endure the least amount of information loss compared to rotations at other angles.

Like the evalution with SNR in Section IV-B2, our model exhibits superior performance compared to painted baseline models, but inferior to the ResNet model as the upper bound for some rotation angles due to the loss of information. The findings conclusively illustrate the GeNet model's proficiency in handling such transformations, showcasing its robustness and adaptability across diverse scenarios without data augmentation.

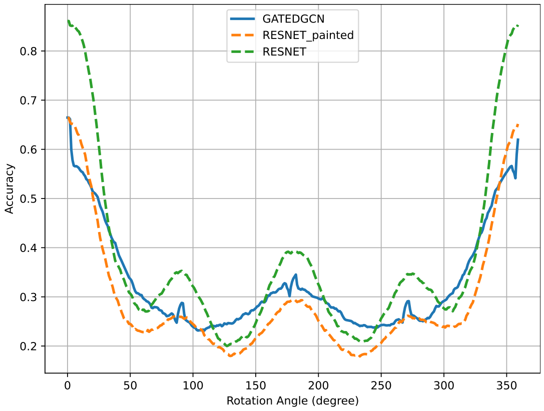


Fig. 10: Test accuracy versus rotation angles of different models in CIFAR10.

V. Conclusion

In this paper, we introduce GeNet, a novel paradigm leveraging Graph Neural Networks (GNNs) for anti-noise, task-oriented semantic communications. Different from traditional methods, GeNet transforms input data images into graph structures and utilizes a GNN-based encoder-decoder model to extract and reconstruct semantic information, respectively. Through extensive experiments on MNIST, FashionMNIST, and CIFAR10 datasets, we have demonstrated that our model is superior performance in anti-noise TOC without prior knowledge of channel conditions, thus saving time and computational resources required for traditional methods trained at different SNRs. Moreover, GeNet can process images of varying resolutions without resizing, ensuring preservation of information integrity, and exhibits robustness to geometric transformations such as rotations without data augmentation. Moving forward, exploring GeNet's potential in transmitting data originally structured as graphs, handling images of different sizes, addressing geometric transformations, and refining graph transformation and noise handling schemes are promising avenues for further research in semantic communication.

References

[1] X. Luo, H.-H. Chen and Q. Guo, “Semantic communications: Overview open issues and future research directions”, IEEE Wireless Commun., vol. 29, no. 1, pp. 210-219, Feb. 2022.

[2] J. Dai, P. Zhang, K. Niu, S. Wang, Z. Si and X. Qin, “Communication beyond transmitting bits: Semantics-guided source and channel coding”, IEEE Wireless Commun., vol. 30, no. 4, pp. 170-177, Aug. 2023.

[3] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton and J. Leskovec, “Graph convolutional neural networks for Web-scale recommender systems”, Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 974-983, Jul. 2018.

[4] J. M. Stokes et al., “A deep learning approach to antibiotic discovery”, Cell, vol. 180, no. 4, pp. 688-702, Feb. 2020.

[5] X. Zhou et al., “Edge computation offloading with content caching in 6G-enabled IoV”, IEEE Trans. Intell. Transp. Syst., Jan. 2023.

[6] X. Zhou, Y. Zhang et al., "Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning", Neural Comput. Appl., vol. 34, no. 7, pp. 5549-5559, Apr 2022.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, May 1998.

[8] H. Xiao, K. Rasul and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms", arXiv preprint, 2017.

[9] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

[10] H. Xie, Z. Qin, G. Y. Li and B.-H. Juang, "Deep learning based semantic communications: An initial investigation", Proc. IEEE Glob. Commun. Conf., pp. 1-6, 2020.

[11] Z. Weng, Z. Qin and X. Tao, "Task-Oriented Semantic Communications for Speech Transmission," 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)., pp. 1-5, 2023.

[12] E. Bourtsoulatze, D. B. Kurka and D. Gunduz, "Deep joint source-channel coding for wireless image transmission", IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 3, pp. 567-579, 2019.

[13] M. Jankowski, D. Gündüz and K. Mikolajczyk, "Deep joint sourcechannel coding for wireless image retrieval", ICASSP 2020-2020 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP). IEEE, pp. 5070-5074, 2020.

[14] J. Shao, Y. Mao and J. Zhang, "Task-oriented communication for multidevice cooperative edge inference", IEEE Trans. Wireless Commun., vol. 22, no. 1, pp. 73-87, Jan. 2023.

[15] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains", IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83-98, May 2013.

[16] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, vol. 1, no. 2, 2017.

[18] H. Zhou, R. Kannan, A. Swami and V. Prasanna, "HTNet: Dynamic WLAN Performance Prediction using Heterogenous Temporal GNN," IEEE INFOCOM 2023 - IEEE Conference on Computer Communications., pp. 1-10, 2023.

[19] X. Zhou, Y. Zhang, Z. Li, X. Wang, J. Zhao and Z. Zhang, "Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning", Neural Comput. Appl., vol. 34, no. 7, pp. 5549-5559, Apr. 2022.

[20] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods", IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[21] P. H. Avelar, A. R. Tavares, T. L. da Silveira, C. R. Jung and L. C. Lamb, "Superpixel image classification with graph attention networks", 2020 33rd SIBGRAPI Conference on Graphics Patterns and Images (SIBGRAPI), pp. 203-209, 2020.

[22] B. Knyazev, G. W. Taylor and M. Amer, "Understanding attention and generalization in graph neural networks", Proc. Adv. Neural Inf. Process. Syst., pp. 4204-4214, 2019.

[23] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," 2020, arXiv:2003.00982.

[24] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, "Neural message passing for quantum chemistry", Proc. 34th Int. Conf. Mach. Learn., pp. 1263-1272, 2017.

[25] A. Paszke et al., "PyTorch: An imperative style high-performance deep learning library", Proc. Int. Conf. Neural Inf. Process. Syst., pp. 8026-8037, 2019.

[26] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, et al., "Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs", CoRR, vol. abs/1909.01315, 2019.

[27] Xavier Bresson and Thomas Laurent, "Residual gated graph convnets," arXiv preprint arXiv:1711.07553, 2017.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 [cs.LG], 2014.

[29] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 770-778, 2016.