STEGANOGRAPHER DETECTION VIA ENHANCEMENT-AWARE GRAPH CONVOLUTIONAL NETWORK

Zhi Zhang[1,†], *Mingjie Zheng*[2,†], *Sheng-hua Zhong*[1,*] *and Yan Liu*[3]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[2]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
zhangzhi2018@email.szu.edu.cn, mingjie.zheng@connect.polyu.hk, csshzhong@szu.edu.cn,
csyliu@comp.polyu.edu.hk

ABSTRACT

Steganographer detection aims to find guilty users who hide secret information in images or other multimedia data in the social network. In existing work, the distances between users are calculated based on the distributions of all images shared by the corresponding users, then users lying an abnormal distance from others are detected as guilty users. This flattened method is difficult to grasp the nuances of the guilty and innocent users. In this paper, we are the first to propose a graph-based deep learning framework for steganographer detection. The proposed Enhancement-aware Graph Convolutional Network (EGCN) represents each user as a weighted complete graph and learns to highlight the differences between guilty users and innocent users based on the structured graph. Compared with the state-of-the-art method and other representative graph-based models, the proposed framework demonstrates its effectiveness across image domains, and even under the context of large-scale social media scenario.

Index Terms— Image steganographer detection, graph convolutional network, graph-based classification

1. INTRODUCTION

Image steganographer detection is a task dedicated to recognizing internet users who may have hidden information in shared images. Different from steganalysis that attempts to catch differences between stego images (embedded with messages) and cover images (without messages), steganographer detection is more concerned about the differences between users. A variety of frameworks have been proposed to set new performance continuously. In 2018, Zheng *et al.* proposed a multi-class deep neural network for steganographer detection task (MDNNSD) [1]. The introduced model achieves state-of-the-art performance on standard datasets.

The existing steganographer detection methods generally consist of two parts: feature extraction and feature-based clustering or outlier detection. Although the feature extraction parts of these methods are different, these methods are almost identical in representing users, measuring differences between users, and detecting suspicious users. In these methods, the distances between users are measured, each of whom is represented by the distributions of images all of his/her own. And then, the guilty user is identified, who deviates too much. The most common case, however, is only a small percentage of images from the guilty users are stego images, and the distribution that is used to represent the guilty user would be very similar to that of the innocent user. Therefore, in steganographer detection task, images lying an abnormal distance from others should be given more significances to represent users.

As we know, despite the difference in content and style, images spread by users always have a relationship with each other, which is crucial for steganographer detection: cover images have commons in containing no embedded message and stego images share similar embedding algorithms or payloads. To represent these relations, it is natural to view the shared images and their relations as a graph, where nodes are images and edges represent relations. Then, geometric methods can be adopted to enhance the common pattern shared by adjacency nodes so that differences between innocent users and guilty users can be highlighted to identify the latter.

In terms of geometric methods, the past few years have witnessed a large number of real-world applications of graph neural networks (GNNs), which try to generalize deep learning methods to non-Euclidean structured data such as graphs and manifolds. The adopted architecture also varies greatly ranging from Recurrent Graph Neural Networks (RGNNs) to Graph Convolutional Networks (GCNs). Encouraged by

This work was supported by the Natural Science Foundation of Guangdong Province (No. 2019A1515011181), the Science and Technology Innovation Commission of Shenzhen under Grant (No. JCYJ20190808162613130), the Shenzhen high-level overseas talents program, the National Engineering Laboratory for Big Data System Computing Technology.

*S.-h. Zhong is the corresponding author of this paper.
†Z. Zhang and M. Zheng contributed equally to this paper.

the success of convolutional neural networks, a large number of GCNs that redefined the notation of convolution for graph data reach the state-of-the-art. In their pilot work, two strategies are used to generalize convolutions on the graph, including spectral-based GCNs [2, 3] and spatial-based GCNs [4, 5, 6]. In these GCNs, approaches inspired by the framework of GCN [3], AGNN [6] and GraphSAGE [4] are currently the most common choices for the graph convolution operation. Inspired by them, we are the first to propose a steganographer detection method via a graph-based convolutional network. To our knowledge, no existing work attempts to explore how to automatically represent and identify steganographer via geometric deep learning methods.

In this paper, we design an alternative to the existing framing that has prevailed in the literature. The novelty of our contributions can be highlighted as: (i) We are the first to propose a geometric deep learning architecture to represent and identify steganographers. (ii) The enhancement-aware graph convolution module is proposed to superimpose and enhance similar features to highlight the differences between guilty users and innocent users. (iii) We carry out extensive experiments to support that our designed graph convolutional network achieves superior performances across spatial and frequency domains.

2. PROPOSED METHOD

In this paper, we design a graph-based deep learning framework to detect guilty users. Fig. 1 illustrates the framework of Enhancement-aware Graph Convolutional Network (EGCN) based Steganographer Detection, which includes two stages: the feature extraction and the graph-based steganographer detection. As we described before, the focus of our paper is to explore how to automatically represent and identify steganographer via geometric deep learning method. Thus, in the feature extraction part, we simply use the deep network introduced by the state-of-the-art method MDNNSD [1] to extract features from each image. For each image from a user, an l-D feature vector is extracted by the network in MDNNSD. These features are fed into the Enhancement-aware Graph Convolutional Network that is proposed to represent and identify the guilty users and innocent users.

2.1. Network Architecture

As shown in Fig. 1, the first step of the EGCN architecture is feature embedding. $F_X = (x_1, ..., x_n)$ is used to denote the original features obtained from user X, where x_i represents an l-D feature vector extracted from an image of user X, and n is the number of images owned by this user. In MDNNSD, l is set to 320. Although extracted feature vectors from images are effective for steganographer detection, the relationship between images is not explored under the representation. To extract high-level features contributing to users' suspicion, a two-layer perceptron is performed to embed original feature vectors into a low-dimensional (80-D) space. Trained as a part of our model, the perceptron can be self-optimized to extract task-related features, minimizing overall framework loss. Here, the ReLU function is used as the activation function. The reduced features for user X can be denoted as $F'_X = (x'_1, ..., x'_n)$. Then, F'_X is fed into two Enhancement-aware Graph Convolutional (EGC) Modules.

Illustrated in Fig. 1, two EGC Modules are used to map the feature vector of each node to lower-dimensional representation with high-level information, where cylinders denote feature vectors and the height of a cylinder represents the dimension of the feature vector. In the EGC Module, we perform graph representation and convolution on user X: graph representation denotes user X as a graph where the i-th node corresponds to feature vector x'_i and graph convolution further transform F'_X into $Z_X = (z_1, ..., z_n)$.

To obtain the representation of a user, we need to aggregate information of feature vectors of shared images and form an effective user representation. From the perspective of graph, it means to aggregate information of nodes to represent the graph. This procedure is usually called the readout or graph coarsening operation. In this paper, we flatten Z_X into a vector Z'_X to represent the user. Compared to other operations like sum, average or max-pooling, this operation remains the feature of the stego images to the utmost extent. Then, we use Z'_X as the input of a one-layer perceptron to classify user X and obtain the classification probability. Here, sigmoid is used as the activation function. Then a two-class classification logistic loss can be optimized. In this way, our method ignores the contingency of a single image and pays attention to whether a user can be classified correctly, which enhances the robustness of our method.

As described before, the core part of our network is the EGC Module. Therefore, we introduce it in detail in Section 2.2.

2.2. Enhancement-aware Graph Convolutional Module

We design the EGC Module to aggregate similar features of shared images and enhance the common pattern from stego images adaptively. The enhancement of common pattern helps to measure users of being guilty and finally identify the steganographer successfully. As shown in Fig. 2, there are two EGC modules in our network, and each module contains two parts: three aggregation blocks and an affine transformation layer. Each block includes a graph representation layer and a graph convolutional layer.

In an aggregation block, user X is represented as a graph where the i-th node corresponds to the feature vector v_i of the i-th image. A common idea about graph representation is constructing unweighted graphs where the nodes only have two states, i.e., connected and unconnected states, and connected nodes in a graph likely belong to the same cluster [3].

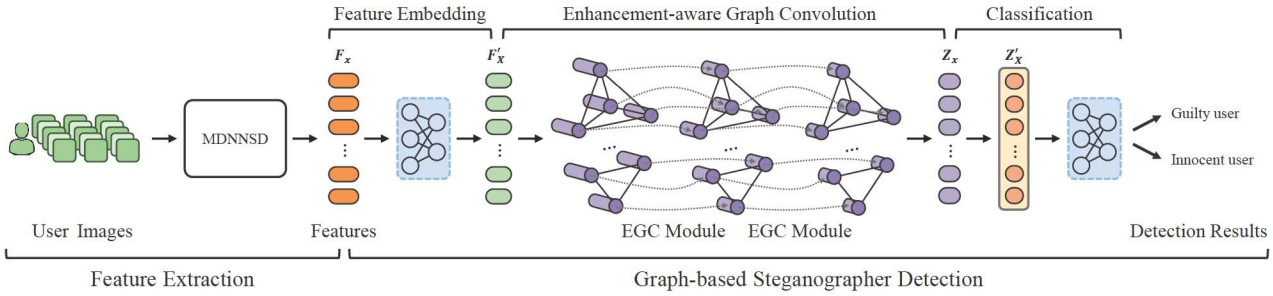

Fig. 1. Enhancement-aware Graph Convolutional network (EGCN) based steganographer detection.

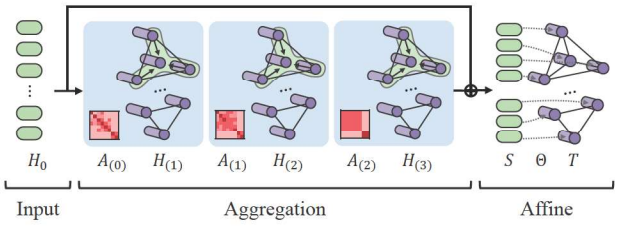

Fig. 2. Enhancement-aware Graph Convolutional Module (EGC Module).

In this task, it is difficult to define a standard to judge whether the nodes in our graph is connected or not. Thus, the graph representation layer uses the weighted complete graph to represent each user. Nodes in a graph are all connected, and a complete graph is formed to retain the relationship between nodes. In detail, we learn from the heat kernel [7] to define the weight of the edge, which is used to describe the similarity between the nodes. The graph for user X can be denoted as an adjacency matrix that is defined as follows:

$$A_{ij} = e^{\frac{-\|v_i - v_j\|_2}{t}} \tag{1}$$

where t is set to 2. Needless to say, the similarity is a number between 0 and 1, where 0 means node i and node j are completely dissimilar while 1 means exactly the same.

In our experiments, the adjacency matrix in Eq. (1) is applied in both of spatial domain and frequency domain. In the experiments on the social network dataset, the graph representation is based on the Cosine similarity rather than Euclidean distance:

$$\begin{aligned} A_{ij} &= \sqrt{e^{S(v_i,v_j)}} \\ S(v_i, v_j) &= \frac{v_i \cdot v_j}{\max(\|v_i\|_2 \cdot \|v_j\|_2, 1e^{-8})} \end{aligned} \tag{2}$$

Guided by the constructed graph structure, the EGC Module is enabled to aggregate information from the graph and enhance common patterns shared by similar nodes. To be specific, each node aggregates features from its neighborhoods and obtains new representation via the weighted average of first-order neighbor nodes, whose weights are described by the weight of each edge. In other words, in the convolutional layer, the aggregated result r_i of node i is:

$$r_i = \sum_{j \in N_i} A_{ij} v_j \tag{3}$$

where N_i denotes the first-order neighbors of node i. We can find that Eq. (3) is equivalent to:

$$R = AV \tag{4}$$

where $V = (v_1, ..., v_n)$ denotes feature vectors of user X and R denotes aggregated features. We can find $A_{ii} = 1$ according to Eq. (1) and Eq. (2), which means each node has a self-loop to contain their own features in the aggregation.

For Eq. (1) or Eq. (2), if node i is very close to others, r_i will be large. Otherwise, r_i will be small. The large dynamic range of R may cause gradient explosion or vashing. So we narrow the range and conduct normalization in the convolutional layer. Then, Eq. (4) can be further formulated as:

$$R = \hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}} V \tag{5}$$

where $\hat{D}$ is a diagonal matrix and $\hat{D}(i,i) = \sum_j A(i,j)$. Then, in each module, the $(k+1)$-th aggregation block can be formulated as:

$$H_{(k+1)} = \tilde{D}_{(k)}^{-\frac{1}{2}} A_{(k)} \tilde{D}_{(k)}^{-\frac{1}{2}} H_{(k)} \tag{6}$$

where $H_{(k)}$ is the output features of k-th block. Especially, $H_{(0)}$ is the input features of the EGC Module. Here, multiple aggregation blocks enhance the common pattern shared by similar feature vectors iteratively and further ease the detection problem. In general, many convolution operations fixed on the original graph may cause the output features over-smoothed [8], and even lead to the nodes from different clusters becoming indistinguishable. To avoid this problem, we use the obtained $H_{(k)}$ to reconstruct the graph. In other words, the adjacency matrix $A_{(k)}$ is updated based on the current features $H_{(k)}$. Then, $H_{(k+1)}$ can be obtained by operating convolution on the reconstructed graph instead of the fixed original graph. Illustrated in Fig. 2, $A_{(0)}$ is the

original adjacency matrix and the brightness of red indicates the weight of each edge. $A_{(1)}$ and $A_{(2)}$ are different reconstructed adjacency matrixes, which can be calculated via $H_{(1)}$ and $H_{(2)}$, respectively. Considering the embedding signals in stego images are weak compared to the image content, this characteristic of our model is critical to achieving good performance in the steganographer detection task.

To alleviate the information lost during multiple aggregation blocks, we construct a shortcut pathway [9] to directly connect the input features $H_{(0)}$ of the module and the output features $H_{(\lambda)}$ of the last block in the module, where λ is the number of aggregation blocks. Mathematically, the shortcut connection can be defined as:

$$S = H_{(\lambda)} + H_{(0)} \tag{7}$$

Finally, the output features of the shortcut connection are fed into an affine transformation layer to transform these features into higher-level features. Mathematically, the affine transformation layer can be defined as follows:

$$T = S\Theta \tag{8}$$

where Θ is the training parameter of an affine transformation layer and T denotes the output features of an EGC Module.

3. EXPERIMENTS

To evaluate our proposed method, a series of empirical evaluations are carried out on two standard datasets and a self-constructed social media dataset. In spatial domain, we run the experiments on the standard dataset BOSSbase ver 1.01 [10]. Then, we extend our method to frequency domain, in which the dataset BOSSbase ver 1.01 compressed with JPEG quality factor (QF) 80 is utilized to evaluate the proposed method. In the end, we investigate the effectiveness of our method under a complex real-world context, where the images from Flickr are collected to construct a new social media dataset. We also provide the implementation details of our proposed method in the supplemental material. The comparisons in the experiments include the state-of-the-art method MDNNSD [1] and the well-known graph-based models: GCN [3], AGNN [6] and GraphSAGE [4]. Notably, there are no existing graph-based models proposed for steganographer detection task. To show the performance gain brought by the proposed core part of EGCN, i.e., EGC Module, we conduct experiments by replacing EGC Module with graph convolution layers from the well-known graph-based methods, including GCN, AGNN and GraphSAGE.

In the proposed network, the number of aggregation blocks λ in each EGC Module is set to 3. In the test stage, the number of users is set to 100, including one guilty user and 99 innocent users. Each user shares 200 images. It is a common setting that there exists likely 0 or 1 guilty user under real scenarios. All the statistical experiments are repeated 100 times, and the average detection accuracies (True Positive Rate) are reported, at which the guilty user is successfully detected from innocent users.

Table 1. Detection accuracies of different methods on S-UNIWARD at different payloads.

Model \ Payload (bpp)		0.05	0.1	0.2	0.3	0.4
State of the art	MDNNSD	4	54	100	100	100
Graph-based	GCN	19	96	100	100	100
	AGNN	24	99	100	100	100
	GraphSAGE	28	88	100	100	100
Proposed	**EGCN**	**82**	**100**	**100**	**100**	**100**

3.1. Steganographer Detection with The Same Steganography

In this experiment, each user randomly samples 200 cover images from the dataset BOSSbase ver 1.01. Innocent users share the original cover images, and guilty users share the stego images embedded secret message at six types of embedding payloads (from 0.5 bpp to 0.05 bpp). In the training stage and the test stage, guilty users use the same steganalytic algorithm, i.e. Spatial version of the UNIversal Wavelet Relative Distortion (S-UNIWARD) [11].

In the training stage, the sequence strategy is used to fit each model from a high payload to a low payload, which is a commonly used training strategy [12] when the payload of test data is assumed unknown. In the test stage, models are tested on five experiments grouped by payloads, including 0.05bpp, 0.1bpp, 0.2bpp, 0.3bpp, and 0.4bpp. The detection accuracy comparisons are shown in Table 1.

As the state-of-the-art steganographer detection method, MDNNSD can perfectly identify the guilty user when the payload is greater than 0.1. But in this paper, we are more concerned about the challenging cases of the steganographer detection, i.e. lower payloads. In this way, we can find most graph-based methods outperform MDNNSD and the successfully designed graph-based model EGCN achieves the best performance when the payload is 0.1 bpp or lower.

3.2. Steganographer Detection against Other Steganography

In this section, we further conduct the sensitivity experiment to test our method when the steganographic algorithms are mismatched. In detail, without retraining models, we directly use the learned models in Section 3.1 for testing. In the test stage, guilty users change to use HIgh-pass Low-pass Low-pass (HILL) [13] as the steganalytic algorithm to embed se-

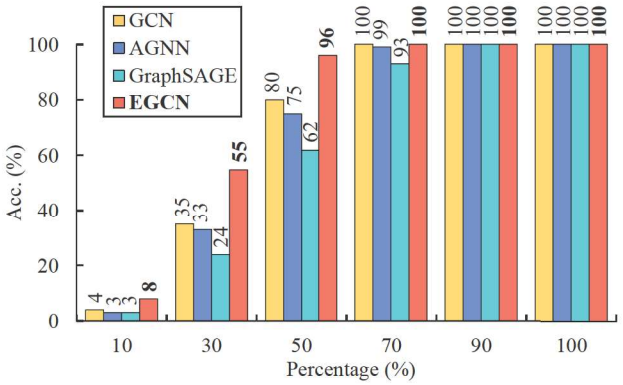

Fig. 3. Detection accuracies of different graph-based methods on HILL at 0.2 bpp when the percentage of stego images ranges from 10% to 100%.

cret messages. Moreover, to further simulate real-life scenarios, guilty users replace some stego images by cover images so that the percentage of the stego images ranges from 10% to 100%. At 0.2 bpp payload, the detection accuracies of graph-based methods are given in Fig. 3.

We can find the detection performance of all methods tends to increase with the increase of the transmitted proportion of stego images. This phenomenon can be easily understood. When the transmitted proportion of stego images by the guilty user is low, most of images shared by the guilty user are normal natural images, i.e. cover images. In this case, the representation of the guilty user would be very similar with that of innocent users. Therefore, it is very difficult to distinguish the guilty user from innocent users. With the help of the graph convolution, the proposed EGCN can automatically superimpose and enhance the similar features to highlight the differences between guilty users and innocent users. It makes the EGCN achieve 55% accuracy rate even in the very difficult case where the transmitted proportion of stego images by guilty users is only 30%. Compared with the sub-optimal method GCN, the accuracy of our method is 20% higher.

3.3. Steganographer Detection in Frequency Domain

In the frequency domain, the JPEG version of BOSSbase ver 1.01 is utilized to evaluate our method. Similar with the setting in spatial domain, all methods are trained by the sequence training strategy when the guilty user utilizes J-UNIWARD at different payloads to embed messages.

As shown in Table 2, the results of different graph-based methods for steganographer detection are provided when the guilty user adopts different JPEG steganographic algorithms to embed messages: J-UNIWARD [11], no-shrinkage F5 (nsF5) [14], and Uniform Embedding Revisited Distortion (UERD) [15]. The payload is set to 0.05, 0.1, 0.2, 0.3, 0.4 bpnzAC, respectively. From these results, we can find all

Table 2. Detection accuracies on J-UNIWARD, nsF5 and UERD at different payloads when the model is trained on J-UNIWARD.

Steganography	Model \ Payload (bpnzAC)	0.05	0.1	0.2	0.3	0.4
J-UNIWARD	GCN	16	88	100	100	100
	AGNN	13	84	100	100	100
	GraphSAGE	13	68	100	100	100
	EGCN	**17**	**90**	**100**	**100**	**100**
nsF5	GCN	24	90	100	100	100
	AGNN	20	90	100	100	100
	GraphSAGE	21	91	100	100	100
	EGCN	**24**	**92**	**100**	**100**	**100**
UERD	GCN	33	96	100	100	100
	AGNN	29	94	100	100	100
	GraphSAGE	25	91	100	100	100
	EGCN	**35**	**97**	**100**	**100**	**100**

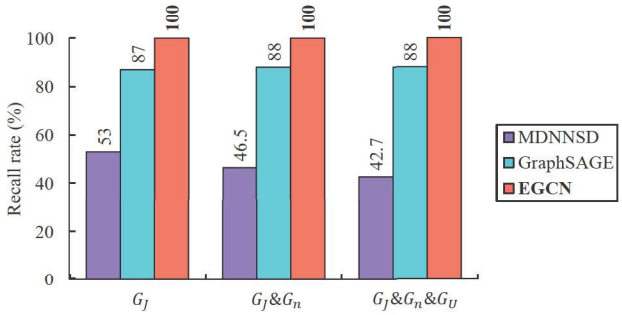

Fig. 4. The average recall rate of different methods in real-world large-scale social network when detecting multiple guilty users.

the methods achieve good performance when the payload is equal to or greater than 0.1 bpnzAC. In particular, our proposed method outperforms others in low payloads. From the principle of embedding framework, J-UNIWARD and UERD are content-adaptive steganographic algorithms while nsF5 is not. From the above results, our method shows the generalization ability whenever the guilty user uses content-adaptive steganography or not.

3.4. Steganographer Detection in Social Network

We run the experiments on a self-constructed social media dataset, in which all images are collected from Flickr. As the steganalytic features are sensitive to the quantization matrices, we compress the collected images with the same quality factor 80. In the experiments, without retraining models, we

directly use the learned model in frequency domain (Section 3.3) to test the generalization ability of our model. In the test stage, each user randomly samples 200 cover images from the self-constructed social media dataset. Innocent users share the original cover images, and guilty users share the stego images embedded secret message at 0.4bpnzAC. There are three kinds of guilty users using three different types of steganalytic algorithms. For the convenience of description, we denote these three kinds of guilty users as G_J, G_n, G_U, whose images, i.e. g_J, g_n, and g_U, are generated by J-UNIWARD, nsF5, and UERD, respectively.

Fig. 4 provides the detection accuracies on this social media dataset when the number of guilty users is equal to 1, 2, and 3, respectively. From these results, we can also find it is easily observed that although the content and the hiding ability of the images in social media dataset varies widely and the proposed model is trained on a different dataset, our method can still obtain quite a good detection results no matter how many guilty users are. To further verify the analyses, we give a visualization experiment on the social media dataset in the supplemental material.

4. CONCLUSION AND FUTURE WORK

In this paper, we are the first to propose a steganographer detection method via graph-based convolutional neural network. Each user is represented as an undirected complete weighted graph and EGCN is performed to enhance the discrimination between the graph of guilty users and that of innocent users. The designed framework can accurately identify the guilty user from easy to hard conditions. By releasing the assumption that all images from the guilty user are stego images, experimental results show that our proposed model performs satisfactorily in the real-world scenario where the guilty user only embeds a small proportion of images. Although our method is designed in spatial domain, it can still achieve much lower detection errors in both spatial and frequency domains. Our model exhibits good generalization ability under the context of large-scale social media that could partly solve the cover-source mismatch problem.

5. REFERENCES

[1] Mingjie Zheng, Sheng-hua Zhong, Songtao Wu, and Jianmin Jiang, "Steganographer detection based on multiclass dilated residual networks," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, 2018, pp. 300–308.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 3844–3852.

[3] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[4] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 1024–1034.

[5] Shikhar Vashishth, Prateek Yadav, Manik Bhandari, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar, "Graph convolutional networks based word embeddings," *arXiv preprint arXiv:1809.04283*, 2018.

[6] Kiran Koshy Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li, "Attention-based graph neural network for semi-supervised learning," *arXiv preprint arXiv:1803.03735*, 2018.

[7] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 15*, 2002, pp. 585–591.

[8] Qimai Li, Zhichao Han, and Xiao-Ming Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI Conference on Artificial Intelligence 32*, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[10] Patrick Bas, Tomáš Filler, and Tomáš Pevný, ""Break our steganographic system": The ins and outs of organizing boss," in *Information Hiding*. Springer, 2011, pp. 59–70.

[11] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1, 2014.

[12] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in *2016 IEEE international conference on image processing*, 2016, pp. 2752–2756.

[13] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing*. IEEE, 2014, pp. 4206–4210.

[14] Jessica Fridrich, Tomáš Pevný, and Jan Kodovskỳ, "Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities," in *Proceedings of the the 9th Workshop on Multimedia & Security, MM&Sec'07*, 2007, pp. 3–14.

[15] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi, "Using statistical image model for jpeg steganography: Uniform embedding revisited," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2669–2680, 2015.