The Application of Graph Attention Mechanism in the Automation of Analog Circuit Design

Xinpeng Li[1], Minglei Tong[1], and Yongqing Sun[2]

[1] EE School, Shanghai University of Electric Power, Shanghai, China PRC
tongminglei@shiep.edu.cn
[2] College of Humanities and Sciences, Nihon University,Chiyoda, Japan
nakahara.eisei@nihon-u.ac.jp

Abstract. Automated annotation of circuit structures can generate hierarchical representations of analog circuit networks, thereby advancing the development of automated analog circuit design tasks. This paper introduces a graph attention network-based model that transforms circuit netlists into graph structures, proposes a feature extraction strategy to learn and predict the circuit structures composed of nodes in the netlists, and presents a method for quickly generating a large number of SPICE circuit netlists to provide ample data for training the graph model. Experiments compared the recognition effects of graph convolutional networks, graph isomorphism networks, and GraphSAGE on the same dataset. The results show that the GAT model outperforms the other models in accuracy, precision, and mean average precision, achieving 90.9%, 91.6%, and 91.9%, respectively. These results demonstrate the superiority of the GAT model in capturing circuit connections, especially in terms of its effectiveness in processing complex circuit diagrams.

Keywords: Automated Annotation · Graph attention networks · Circuit structure · Feature extraction

1 Introduction

Analog IC layout automation is one of the key research directions in the field of integrated circuit design, with the goal of fully automating the design of Analog circuits [14]. Existing automated design methods are seldom adopted on a large scale due to the high complexity of Analog circuits' typologies and their variants [7,18], and in previous attempts to solve the problem, matching circuits to pre-specified templates involves traversing all the possible topologies in a database [19,20] , and identification through fixed circuit rules requires rule declarations by experienced researchers [6, 24] , so it is difficult to comprehensively cover all topological variants. Graph neural networks [22] have attracted much attention due to their excellent performance in identifying target variants [4,25] . The application of graph neural networks has also been demonstrated in the field of electronic design automation (EDA). The pins of components in a circuit can be modelled as vertices in circuit design [26], but the isomorphism of the

graph can lead to an excessively large set of vertices, which in turn causes loss of information [5], and defining components and network nodes as disjoint two-part sets [11] reduces the number of vertices while increasing the complexity of detecting adjacent nodes. Further, the development of graph convolution networks can classify circuit components by performing sub-graph isomorphism [2,8], and related studies have demonstrated that graphical representation of circuit netlist followed by graph convolution networks training can achieve circuit classification [15], Analog circuit structure identification [9,10,12], mixed digital-Analog circuit constraints extraction [1,16], layout parasitic parameters prediction [21] , and Analog circuit layout constraint annotation [3] .

In this study, we propose a structural annotation framework based on graph attention networks, which, after converting circuit netlist into graph structures, learns the structural features of these graphs by introducing an attention mechanism, which in turn enables us to dynamically focus on the relative importance between nodes in the graphs, and thus identify critical circuit elements and connections more accurately. The main contributions of this paper can be summarised as follows:

(1) An efficient method is proposed to convert Analog circuit netlist into graph representations, laying the foundation for subsequent graph learning tasks.

(2) A graph attention mechanism is introduced to identify and emphasise the role of important nodes in the circuit diagram, which further improves the accuracy and robustness of the model in recognising circuit structures.

2 Related Work

2.1 Circuit netlist representation and structural recognition

B. Xu et al. abstracted the circuit netlist into a graphical representation in order to achieve constraint extraction, preserving the pin information in the representation, where devices, pins and networks are represented as nodes, pins are connected to the circuit elements they belong to, and networks are connected to the pins according to the circuit netlist's connection to the pins [26] . All edges in this representation are not directly connected to circuit elements, but are first connected to edges. This representation makes the set of circuit vertices obtained too large and there is a risk of information loss.

W. Hamilton et al. used node feature information to generate node embedding for previously unseen data avoided the drawback of large vertex set by generating embedding by learning to sample and aggregate features from the local neighbourhood of the node [5] .K. Kunal et al. reduced the vertex set by defining component vertices and nodes in the circuit netlist as a bipartite set that does not intersect [11]. Using a trained graph convolution neural network to identify netlist elements for designing hierarchical circuit blocks, the study first created a graph representation of a flat network table based on a custom dataset, and then used graph isomorphism techniques to determine the structure of lower hierarchical structures, and then used the graph convolution network to

identify sub-circuits in order to achieve approximate sub-graph isomorphism [9] .

Graph Convolution Networks are highly dependent on the graphical representation and feature engineering of circuits [27] during model design, which may limit the ability of the model to generalise to different types or novel circuit designs, and different circuit structures may be isomorphic in graphical representation but functionally different [17] . This isomorphism may lead to ambiguity in the labelling process of GCNs, affecting the accuracy and reliability of the labelling.

2.2 Graph Attention Networks

Graph Attention Network (GAT) [23] is a neural network structure proposed by Veličković et al. It overcomes some of the limitations of graph neural networks through the mechanism of attention.GAT aggregates the neighbourhood information of a node by assigning different attention weights to the neighbouring nodes of each node in the graph, thus assigning different importance to each node's neighbours in the graph [13] , which improves the predictive and generalisation capabilities of the network . The attention mechanism is defined as shown in Equation 1.

$$\text{Attention}\left(\text{Query}, \text{Source}\right) = \sum_{i} \text{similarity}\left(\text{Query}, \text{Key}_i\right) \cdot \text{Value}_i \tag{1}$$

Source refers to the data source that needs to be processed, *Query* represents the prior information, and $similarity(Query, Key_i)$ indicates the correlation between the *Query* vector and the Key vector. *Attention* refers to the process of extracting information from the "Source" based on the condition of the *Query* using the attention mechanism. The various pieces of information in the "Source" are represented in a $Key - Value$ format. Finally, all the $Value$ information is weighted and summed, with the weights determined by the correlation between the *Query* vector and the Key vector.

3 Works

3.1 Graphical Representation of Netlists

Data Preprocessing In this paper, the SPICE format has been chosen as the standard format for circuit netlist data because it is widely used in the field of electronic circuit design and provides a wealth of information to describe the physical properties and connectivity relationships of circuits. Directly analysing the original SPICE netlist faces certain challenges, especially the complexity caused by the hierarchical design of circuits, in order to solve this problem, this paper adopts the pre-processing step of flattening processing to simplify the structure of the circuit netlist.

The core aim of the flattening process is to transform the hierarchical circuit netlist into a form that is simpler in structure and easier to analyse. By eliminating the hierarchical structure of the circuit and flattening all components and their connections to the same level, it provides an intuitive and unified view of the circuit representation. This transformation not only simplifies the data analysis and processing process, but also facilitates the implementation and application of circuit rules.

Following the flattening of the circuit netlist, the next stage involves a process of converting it into a graph structure, which involves extracting components from the netlist as nodes of the graph, and considering the electrical connections between the components as edges of the graph, thereby constructing a graph representation reflecting the topology of the circuit. Further, feature vectors and adjacency matrices are extracted from this graph structure to build the necessary data structure base for the inputs to the graph neural network model. The feature vector carries descriptive information about each node (i.e., circuit element), while the adjacency matrix reveals the connection patterns between nodes. The combined use of these two data structures provides a rich set of information resources for the graph attention network model, which enables the model to effectively learn the structural properties of the circuit and improves the accuracy and efficiency of the analysis.

Circuit Netlist Generation The lack of datasets is an important factor affecting the development of Analog circuit design automation, this section focuses on the generation process of custom datasets, for the structural annotation of Analog circuit netlists, this paper proposes a hybrid netlist generation method based on circuit rules, and the flow is shown in Fig 1. The method is mainly

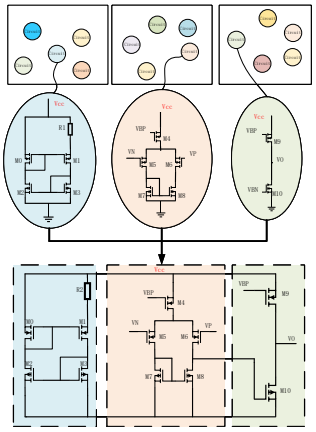

Fig. 1. Schematic diagram of circuit netlist generation process

Table 1. Naming Constraints of Circuit Netlists.

Naming Element	Description
M	MOSFET
R	Resistor
C	Capacitor
Vcc	Power
Gnd	Ground

used to automate the generation of circuit netlists, especially for the comprehensive design of operational amplifiers and bias circuits. The method adopts an innovative algorithmic flow to automatically read the SPICE format sub-circuit netlist files stored in a specified directory, which are designed based on certain circuit rules with the same naming constraints rules, e.g., the output pins of a bias circuit need to be connected to the bias input pins of the operational amplifier, and by applying these matching rules, the program dynamically generates connection By applying these matching rules, the program dynamically generates connection commands to automatically connect the corresponding pins together, eliminating the hierarchical structure of the circuit and flattening all components and their connections to the same level, simplifying the structure of the circuit netlist, thus making the subsequent analysis more efficient and direct. The naming constraints of the netlist are shown in Table 1.

Graphical Representation of Circuit Netlists In this paper, we use an undirected bipartite graph G(V,E) to represent a network table of circuits, and all the nodes in the network table as a set of vertices, which are divided into two categories: components and electrical nodes. The set E consists of edges between nodes. Figure 2 shows an example of a graphical representation of a circuit netlist. The circuit diagram consists of two MOS tubes and multiple electrical

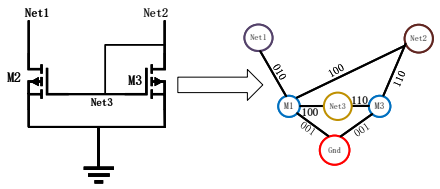

Fig. 2. Graphical representation of circuit netlist

nodes, the electrical nodes and components are set as nodes, in which the purple, yellow and red nodes represent different electrical nodes, respectively, and different features are assigned during feature extraction, and the blue vertices

represent the components, and when the graph representation is carried out, in order to be able to show the connection relationship between transistors and the surrounding nodes more clearly, the connection to the transistors is provided. Each edge is assigned a three-bit label, which represents the connection relationship between the three stages of the transistor and its neighbouring nodes in the order of lgldls, as shown in figure 2, the leakage stage of M2 is connected to the Net1 node, so the label of the edge between M2 and Net1 is 010, and these labels will be used as a part of the node feature matrix in the subsequent node feature extraction.

Node Feature Extraction The node features in this context are represented in the form of an adjacency matrix and an n×d feature matrix, where d represents the number of features. To better capture the overall structural characteristics of the circuit netlist, each node's features are associated with a 22-dimensional feature vector, as outlined below:

- 5 features are related to the type of component that the node represents.
- 5 features correspond to the type of electrical node the component is connected to.
- 3 features describe the circuit structure in which the node is located.
- 9 features are linked to the labels of the edges connected to the node.

3.2 Graph Attention Model Construction

The Graph Attention Model proposed in this paper is composed of three stages: graphical representation of the netlist, feature extractor, and predictor, as shown in Figure 3. The feature extractor consists of two layers of Graph Attention Networks (GAT), with each layer comprising graph convolution and graph pooling operations. The predictor is a fully connected neural network, also known as a Multi-Layer Perceptron (MLP).

Node Feature Extraction In order to better represent the circuit netlist through the graph and avoid circuits with different topologies having homogeneous graph representation, so for different nodes within the netlist to construct their special node features, based on the feature extraction rules in 3.1 , the global attributes of the node are added on top of it to indicate whether it belongs to more than one sub-circuit at the same time as well as whether it is a global signalling node or not, which will be comprehensively represented into the the feature matrix.

GAT Node feature aggregation is implemented using a two-layer GAT network, where for each node, GAT first calculates the attention coefficient between that node and each of its neighbours through a learnable weight matrix and an attention mechanism that calculates the attention coefficient based on the features of the current node and its neighbours. The input to the GAT network in this paper is a set of circuit netlist node feature matrices:

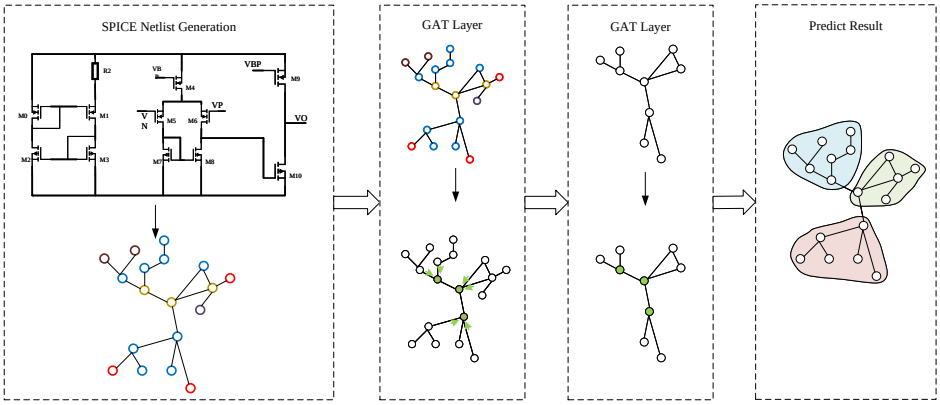

Fig. 3. Graph attention network for circuit structure recognition

$$X = \left[\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_m}\right] \tag{2}$$

After input to the GAT model, the attention coefficients are calculated by node features$\overrightarrow{h_i}$ and neighbour node features $\overrightarrow{h_j}$:

$$e_{ij} = a^T Leaky\mathrm{ReLU}\left(\left[\overrightarrow{h_i} \middle\| \overrightarrow{h_j}\right] W^T\right) \tag{3}$$

Where W is a linear transformation applied to the features of each node to obtain more expressive feature vectors, e_{ij} is the computed attention coefficient, representing the importance of neighboring node j to node i, and a^T is the attention weight matrix. The attention coefficients are normalized using the softmax function, yielding:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in N_i} \exp\left(e_{ik}\right)} \tag{4}$$

Where k is the information of neighbour nodes other than neighbour node j. The normalized attention coefficient α_{ij} is obtained by equation (3) . For the ith node feature in the feature matrix X, the output of the GAT layer is.

$$\overrightarrow{h_i'} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W \overrightarrow{h_j}\right) \tag{5}$$

Where $\overrightarrow{h_i'}$ represents the output feature of node i, N_i represents all neighboring nodes of node i, and σ is the activation function. Through the GAT , the features of neighboring nodes are further learned, resulting in the output feature matrix X'.

$$X' = \left[\overrightarrow{h_1'} \, \overrightarrow{h_2'} \cdots \overrightarrow{h_m'} \right] \tag{6}$$

Through the above method, the attention mechanism is introduced to dynamically learn the interaction weights between nodes, which is more flexible in dealing with graph structure data, captures the complex dependencies between nodes, and is able to adapt to different graph structures, which improves the adaptability and prediction accuracy of the model to novel circuit designs to a certain extent.

Result Prediction A fully connected layer is set up after the output of the GAT layer for predicting the output of the model, along with the nodes and edges contained in each circuit structure.

4 Experimental Results and Analysis

4.1 Experimental Environment

The experimental environment used in this paper is as follows:

- GPU: RTX 3060
- CPU: Core i5-12600KF
- Memory: 16GB
- Python: 3.8.17
- PYG (PyTorch Geometric): 2.3.1
- PyTorch: 2.0.0
- CUDA: 11.6

4.2 Evaluation Metrics

In the detection results, TP (True Positive) indicates the number of correctly predicted subcircuit nodes, i.e., the predicted nodes belong to the subcircuit and are predicted correctly, FP (False Positive) indicates that the predicted nodes belong to the subcircuit but are predicted incorrectly, TN (True negative), indicates that the predicted nodes do not belong to the subcircuit and are predicted correctly, and FN (False Negative), which indicates that the predicted node does not belong to the subcircuit, but the prediction is wrong.

Accuracy Accuracy is one of the most common evaluation criteria, which indicates the ratio of the number of correctly classified nodes to the total number of nodes, and is usually applied to the case of balanced category distribution. The calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Table 2. Custom Dataset Parameters.

Circuits	Nodes	Edges	Features
1480	44962	135460	22

Precision The proportion of correct prediction results for all nodes belonging to a subcircuit is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

mean Average Precision AP is the average precision of the prediction results of each subcircuit. mAP calculates the average of the average precision AP of the detection results of all categories of subcircuits with the following formula:

$$AP = \int_0^1 P(R)dR \tag{9}$$

$$mAP = \sum_{n-1}^{N} AP\,(n)/N \tag{10}$$

The formula states that $P(R)$ represents precision as a function of recall, and $AP(n)$ denotes the average precision for the nth query.

4.3 Dataset Preparation

The circuit netlist data converted to graphical representation is encapsulated through the Torch_Geometric library, and the connection matrix A is encapsulated in the DATA data along with the feature matrix X, which is saved in the local directory in the form of a pt-file, and the rules for obtaining the connection matrix A and the feature matrix X are described in detail in 3.1 , and the main parameters of the customised dataset in this paper are shown in Table 2.

4.4 Experimental Results

In this experiment, structural labelling experiments have been conducted on a custom dataset, and comparative experiments have been conducted for several existing graph models in order to test the effectiveness of the graph attention mechanism in the field of structural labelling of Analog circuits.

The experiment divides the samples into training set, validation set, and test set in the ratio of 7:2:1, which are independent of each other, and uses two layers of GAT for training.Through the application of two consecutive attention mechanisms, we are able to consider the neighbours of nodes as well as the neighbours of neighbours, so as to capture a wider range of information about the graph structure, to enhance the model's understanding of the global structure of graphs, and to improve the ability of feature representation.The size of

Table 3. Experimental Results.

Method	Accuracy	Precision	mAP
GCN	0.893	0.906	0.908
GIN	0.895	0.902	0.909
GraphSAGE	0.902	0.915	0.918
GAT	**0.909**	**0.916**	**0.919**

the input dimension of the GAT layer is 22 dimensions, the number of hidden layer channels is 16 dimensions, using Adam optimiser and binary cross-entropy loss function, configuring weight_decay in the optimiser, using Dropout layer to randomly ignore a group of neurons to prevent overfitting, and after testing it is learnt that the performance of using ReLU activation function is better, a total of 500 epochs are trained, and the optimal results are obtained in the validation set when the Save model parameters.

In order to measure the performance of graph attention network in labelling circuit structures, it is compared with other graph neural network algorithms and the results obtained are shown in Table 3.

As shown in Figure 4 is the circuit diagram of this design after importing into Cadence via the circuit netlist generated in section 3.1 and after manual placement, the actual structure of the circuit is a fully differential amplifier, which is used as an input to output the three sub-structures of the circuit via the GAT network. From the above experimental results, it can be seen that the

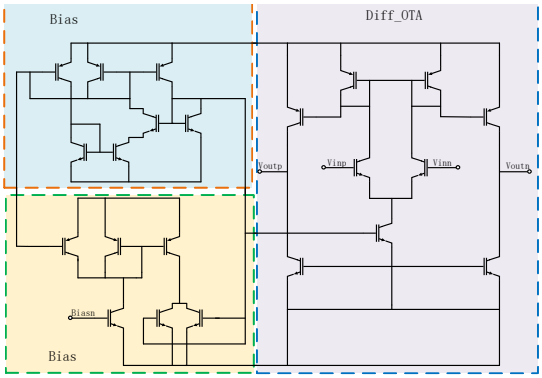

Fig. 4. Circuit structure recognition results

introduction of the graph attention network makes the accuracy of structure recognition improve on each of the other graph models, suggesting that the complex relationships between nodes can be captured more effectively through the use of the graph attention mechanism.

5 Conclusion

To address the challenge posed by the limited availability of large-scale circuit netlists for training and the recognition of analog circuit structures, this paper presents a method for the rapid generation of integrated circuit netlists. Additionally, a Graph Attention Network (GAT) is introduced to facilitate circuit structure recognition. Experimental results demonstrate that the GAT exhibits superior performance and generalization capabilities when processing circuit topologies, effectively capturing critical structural information. This leads to improved accuracy and adaptability in graph classification tasks. Future work may focus on expanding the variety of circuits and incorporating actual circuit component parameters for constraint annotation, thereby enabling fully automated analog layout design.

References

1. Chen, H., Zhu, K., Liu, M., Tang, X., Sun, N., Pan, D.Z.: Universal symmetry constraint extraction for analog and mixed-signal circuits with graph neural networks. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 1243–1248 (2021). https://doi.org/10.1109/DAC18074.2021.9586211 2
2. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering (2017), https://arxiv.org/abs/1606.09375 2
3. Gao, X., Deng, C., Liu, M., Zhang, Z., Pan, D.Z., Lin, Y.: Layout symmetry annotation for analog circuits with graph neural networks. In: 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC). pp. 152–157 (2021) 2
4. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications (2018), https://arxiv.org/abs/1709.05584 1
5. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Neural Information Processing Systems (2017), https://api.semanticscholar.org/CorpusID:4755450 2
6. Harjani, R., Rutenbar, R., Carley, L.: A prototype framework for knowledge-based analog circuit synthesis. In: 24th ACM/IEEE Design Automation Conference. pp. 42–49 (1987). https://doi.org/10.1145/37888.37894 1
7. Huang, G., Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., Wu, J., Xu, Y., Zhang, H., Zhong, K., Ning, X., Ma, Y., Yang, H., Yu, B., Yang, H., Wang, Y.: Machine learning for electronic design automation: A survey (2021), https://arxiv.org/abs/2102.03357 1
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017), https://arxiv.org/abs/1609.02907 2
9. Kunal, K., Dhar, T., Madhusudan, M., Poojary, J., Sharma, A., Xu, W., Burns, S.M., Hu, J., Harjani, R., Sapatnekar, S.S.: Gana: Graph convolutional network based automated netlist annotation for analog circuits. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 55–60 (2020). https://doi.org/10.23919/DATE48585.2020.9116329 2, 3
10. Kunal, K., Madhusudan, M., Sharma, A.K., Xu, W., Burns, S.M., Harjani, R., Hu, J., Kirkpatrick, D.A., Sapatnekar, S.S.: Invited: Align – open-source analog layout automation from the ground up. In: 2019 56th ACM/IEEE Design Automation Conference (DAC). pp. 1–4 (2019) 2

11. Kunal, K., Poojary, J., Dhar, T., Madhusudan, M., Harjani, R., Sapatnekar, S.S.: A general approach for identifying hierarchical symmetry constraints for analog circuit layout (2020), https://arxiv.org/abs/2010.00051 2
12. Li, H., Jiao, F., Doboli, A.: Analog circuit topological feature extraction with unsupervised learning of new sub-structures. In: 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1509–1512 (2016) 2
13. Li, Q., Wang, D., Feng, S., Niu, C., Zhang, Y.: Global graph attention embedding network for relation prediction in knowledge graphs. IEEE Transactions on Neural Networks and Learning Systems **33**, 6712–6725 (2021), https://api.semanticscholar.org/CorpusID:235412138 3
14. Lin, Y., Gao, X., Zhang, H., Wang, R., Huang, R.: Intelligent and interactive analog layout design automation. In: 2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology (ICSICT). pp. 1–4 (2022). https://doi.org/10.1109/ICSICT55466.2022.9963217 1
15. Liou, G.H., Wang, S.H., Su, Y.Y., Lin, M.P.H.: Classifying analog and digital circuits with machine learning techniques toward mixed-signal design automation. In: 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD). pp. 173–176 (2018). https://doi.org/10.1109/SMACD.2018.8434884 2
16. Liu, M., Li, W., Zhu, K., Xu, B., Lin, Y., Shen, L., Tang, X., Sun, N., Pan, D.Z.: S3det: Detecting system symmetry constraints for analog circuits with graph similarity. In: 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC). pp. 193–198 (2020). https://doi.org/10.1109/ASP-DAC47756.2020.9045109 2
17. Liu, M., Zhu, K., Gu, J., Shen, L., Tang, X., Sun, N., Pan, D.Z.: Towards decrypting the art of analog layout: Placement quality prediction via transfer learning. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 496–501 (2020). https://doi.org/10.23919/DATE48585.2020.9116330 3
18. Lopera, D.S., Servadei, L., Kiprit, G.N., Hazra, S., Wille, R., Ecker, W.: A survey of graph neural networks for electronic design automation. 2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD) pp. 1–6 (2021), https://api.semanticscholar.org/CorpusID:237427133 1
19. Massier, T., Graeb, H., Schlichtmann, U.: The sizing rules method for cmos and bipolar analog integrated circuit synthesis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **27**(12), 2209–2222 (2008). https://doi.org/10.1109/TCAD.2008.2006143 1
20. Meissner, M., Hedrich, L.: Feats: Framework for explorative analog topology synthesis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **34**(2), 213–226 (2015). https://doi.org/10.1109/TCAD.2014.2376987 1
21. Ren, H., Kokai, G.F., Turner, W.J., Ku, T.: Paragraph: Layout parasitics and device parameter prediction using graph neural networks. 2020 57th ACM/IEEE Design Automation Conference (DAC) pp. 1–6 (2020), https://api.semanticscholar.org/CorpusID:221679424 2
22. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2009). https://doi.org/10.1109/TNN.2008.2005605 1
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018), https://arxiv.org/abs/1710.10903 3
24. Wu, P.H., Lin, M.P.H., Chen, T.C., Yeh, C.F., Li, X., Ho, T.Y.: A novel analog physical synthesis methodology integrating existent design expertise. IEEE Trans-

actions on Computer-Aided Design of Integrated Circuits and Systems **34**, 199–212 (2015), https://api.semanticscholar.org/CorpusID:14927270 1

25. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386 1
26. Xu, B., Zhu, K., Liu, M., Lin, Y., Li, S., Tang, X., Sun, N., Pan, D.Z.: Magical: Toward fully automated analog ic layout leveraging human and machine intelligence: Invited paper. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–8 (2019). https://doi.org/10.1109/ICCAD45719.2019.8942060 1, 2
27. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018), https://api.semanticscholar.org/CorpusID:46949657 3