Global Routing Under a Congestion-Aware Reinforcement Learning Model

1st Lin Li
Department of Computer Science and Technology
Tsinghua University
Beijing, China
lil19@mails.tsinghua.edu.cn

2nd Yici Cai
Department of Computer Science and Technology
Tsinghua University
Beijing, China
caiyc@tsinghua.edu.cn

3rd Qiang Zhou
Department of Computer Science and Technology
Tsinghua University
Beijing, China
zhouqiang@tsinghua.edu.cn

2023 International Symposium of Electronics Design Automation (ISEDA) | 979-8-3503-0451-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISEDA59274.2023.10218371

***Abstract*—As one of the challenging problems in VLSI physical design, global routing is facing increasing difficulties and more and more algorithms attempt to introduce machine learning-based solutions. While most of these solutions lack high enough routability and routing efficiency. In this paper, we propose a congestion-aware reinforcement learning model for global routing. The global routing problem is solved based on pattern routing and a reinforcement learning technique is introduced for better searching global optimal routing decisions. We convert the network relation into a graph structure and introduce a graph attention network to provide global congestion features. Moreover, a net segment-based feature extraction mode for pattern routing is proposed to enhance the routing efficiency. Experimental results on ISPD18 benchmark show that the proposed algorithm can route different unseen layouts with an average routability over 99.9% and the average number of the failed nets is around 6.**

***Index Terms*—VLSI, Global Routing, Deep Reinforcement Learning, Graph Attention Network.**

I. Introduction

Global routing is a significant stage in physical design. It connects all pins related to the same network within a divided layout. Due to the nodes of a semiconductor scaling down, global routing is now facing increasing challenges. Besides, many traditional global routing methods lack learning ability. In most of the existing studies, routing models cannot learn anything beyond what the expert has designed, which is not flexible enough. It may be essential to search for a more intelligent solution.

Recently, more and more problems in physical design are searching for automated solutions based on machine learning. One popular application is to adopt machine learning to help early prediction, therefore, some preventive measures can be activated to obtain better results. [1] introduces conditional generative adversarial nets to forecast routing congestion for FPGA. [2]–[5] apply machine learning to predict DRC violations in the early stage. [6], [7] predict violations and optimize placement based on the predicted results. In addition, the powerful learning ability of machine learning also inspires experts to search for further automation. [8] first explores the application of artificial intelligence in solving placement with the attempt to ease the difficulties of manual effort, which may indicate a new development stage for physical design.

For global routing, several machine learning-based algorithms are also proposed. [9] first attempts at routing a circuit layout net with a convolutional neural network. Later in 2019, [10] formulates and solves global routing as a deep reinforcement learning problem. [11] proposes a data-independent reinforcement learning-based routing model to route a circuit and correct short violations. In 2020, [12] redefines global routing as a classical image-to-image processing problem and proposes a deep learning system with a variational autoencoder and custom loss function. However, there are still some remaining issues to be solved.

Firstly, most of the existing machine learning-based global routers are difficult to route real circuit designs and lack high routability. [10] experiments on the generated dataset with a net number of no more than 50. [12] evaluates its router on parts of the nets from a public benchmark layout and achieves 96.8% of routability. Besides, the existing machine learning-based global routers may lack routing efficiency. A deep neural network is used to solve global routing in [12], while the input size of each tile is specific and it seems that the router can only route two- and three-pin nets, which may have some limitations for application. [10] introduces deep reinforcement learning to construct a global router. However, the proposed router may suffer from overfitting and it may be difficult to route layouts with different routing settings. Also, the model searches for the global optimal solution iteratively, which is not efficient enough.

In this paper, we propose a congestion-aware reinforcement learning model for global routing. To achieve high routability, we integrate reinforcement learning with pattern routing and apply reinforcement learning to search for optimal pattern decisions. Moreover, a graph attention network is introduced to provide congestion prediction features. To further improve the routing efficiency, we propose a net segment mode for a more clustered and reliable feature extraction. The contributions of this paper are summarized as follows:

- Proposing a congestion-aware reinforcement learning model for global routing and achieving an average routability over 99.9% and 100% routability in many cases.
- Integrating pattern routing with reinforcement learning and searching for optimal pattern decisions.
- Converting the network relation into graph structures and proposing a graph attention network(GAT)-based congestion predictor to provide global congestion features.
- Proposing a net segment mode for pattern routing to enhance the efficiency of feature extraction, which can obtain more clustered and reliable features.

II. Preliminary

A. Global Routing

Global routing plans the connection path based on the placement result. Before global routing, the layout is generally divided into multiple routing areas, also known as GCELL, which is composed of multiple horizontal and vertical grid lines according to the setting of the routing file. Each routing area contains a certain number of routing tracks, and the number of these routing tracks determines the number of nets can be contained in this area, that is, the routing capacity. Global routing should determine the topological routing path of all the nets in the routing area.

B. Pattern Routing

Pattern routing is a commonly used method to solve global routing problems. It routes two-terminal nets by prespecified patterns [13]. Although maze routing, such as A* search algorithm can guarantee to find out the shortest path between two points, these techniques can be time-consuming, especially when the topology of a net consists of only edges with very few vias. In this case, pattern routing is able to help speed up the routing process by generating 2D topologies in a short time to pattern route a large number of nets. Usually, there are two simple patterns including L-shaped with a single bend and Z-shaped with two bends. In this paper, we mainly use L-shaped to integrate with the deep reinforcement learning model. As shown in Figure 1, there are two types of L-shaped patterns.

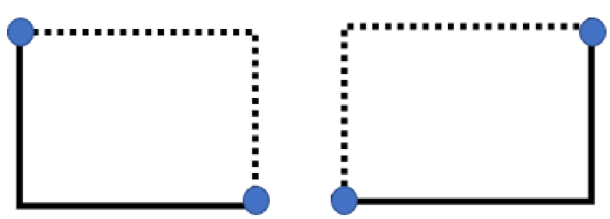


Fig. 1. Two types of L-shaped pattern routing.

C. Deep Reinforcement Learning

Reinforcement learning (RL) is one of the most popular research areas in recent years. It is concerned with training an intelligent agent to achieve a specific goal in an uncertain, potentially complex environment aiming to maximize the cumulative reward. It usually includes several essential elements, such as state space, action space, and reward function. Deep reinforcement learning (DRL) integrates deep learning techniques with reinforcement learning through the use of deep learning to fit the strategy function.

D. Graph Attention Network

Graph attention network(GAT) [16] is a spatial graph neural network with graph data as input. GAT is a model to consider graph structure from the perspective of space, which considers the geometric relation between the target node and other nodes, such as whether there is a connection. It uses an adjacency matrix as a mask and it is characterized by the addition of a graph attention layer, so that it can calculate the importance of each node relative to its adjacent nodes.

III. GAT-based congestion predictor

Congestion information is important to routing and it helps the router to avoid some potential resource shortage areas. To achieve high routability, we introduce GAT to predict congestion.

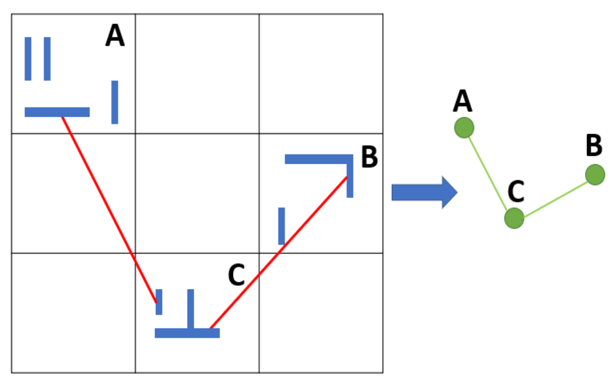


Fig. 2. An example of our graph structure.

Congestion usually occurs where routing resources are insufficient, that is, when there are too many nets passing by. Therefore, network relation is significant for predicting congestion. Before global routing, we can obtain fly lines by straightly connecting pins that are related to the same network, which is one of the most simple methods to present network relations. However, forcibly converting fly lines to Euclidean structures may lead to the loss of some information. Consider that the structure of a fly line is similar to a graph structure. Therefore, we introduce a graph neural network. As presented in Figure 2, each GCELL is regarded as a node, and the fly lines between GCELLs are regarded as the edge. We use edges between nodes to reflect the network relation between GCELLs, while a GCELL may connect to multiple different

GCELLs, and these different GCELLs have different impacts on its congestion. For example, GCELLs that are farther away may have less impact. Therefore, to better deal with this issue, the graph attention network(GAT) is introduced.

We apply GAT to predict the congestion of each GCELL. For each GCELL, we construct a node-set that collects the first-order and second-order connection nodes. The first-order connection nodes are nodes that are directly connected to the current node and the second-order nodes are nodes that are connected to the first-order nodes. The adjacency matrix is constructed based on the node-set. In addition, we extract features for each node to enhance the performance. The extracted features are listed below:

- Node embedding. The effective embedding information obtained from the graph data to assist the expression of node features can effectively improve the prediction 这篇之前没读懂 based on graph neural network [15]. Given the adjacency matrix, we form the embedding based on the Infinite Walk [17] matrix factorization method.
- Pin number. This feature includes the number of pins inside the current GCELL and the number of pins among the neighborhood GCELL.
- Fly line number. The number of fly lines emanating from the current GCELL. 为什么capacity在GCell上
- Capacity value. The capacity value of current GCELL. We consider that the original capacity value has an important effect on congestion. With the same number of routing nets, GCELLs with more capacity are obviously less congested. Besides, capacity values in different layouts are usually different. This feature helps the model predict congestion based on the current capacity and routing environment.
- Bounding box number. A bounding box is the minimum box that covers all pins within the same network. This feature includes the number of bounding box that associated with the current GCELL and the number of bounding box that associated with the neighborhood GCELL. We consider that a network has a high probability to route within its own bounding box, therefore, the number of bounding boxes can reflect the degree of future congestion to some extent.
- Position correlation. This feature calculates the Manhattan distance between the farthest position on the layout and the current GCELL. The position of GCELL also has an impact on congestion. For example, GCELLs near the center of the layout may be more likely to be congested than GCELLs near the margin.感觉有点胡扯

The degree of congestion of a GCELL is not only related to its own features, but also to the features of its neighborhood. Therefore, to enhance the quality of our predictor, we consider both features of the current GCELL and features associated with the current GCELL when exacting node features as mentioned above.

IV. DRL-based Pattern Routing

A. asynchronous advantage actor-critic method

Reinforcement learning obtains information and feedback from the environment to continuously optimize the model. Therefore, complex environments tend to lead to slow convergence due to the expensive query to the environment. To better handle this issue, an asynchronous advantage actor-critic(A3C) method [14] is proposed and it is adopted as our basic DRL model. A3C contains multiple actor-critic(AC) agents and each agent has its policy and value network. Different agents run in parallel and each of them executes actions in its environment to explore the solution space under a different policy. Besides, A3C has its global network and different agents update this network during training, which can help improve the training efficiency. Furthermore, A3C is an n-step method. After each agent executes n steps, the total return and correlation gradient of each step are calculated backwards, and the main network parameters are updated with the cumulative gradient. This mechanism can help the model search for globally optimal solutions, instead of limiting it to the current step.

There are two models in the A3C framework, a policy network and a value network. The policy network outputs the scores of different pattern routing decisions, and the value network evaluates the scores considering future rewards. We set the policy network as a fully connected layer with 200 neurons and the value network as a fully connected layer with 100 neurons.

B. Net segment-based feature extraction

Our global routing problem is transformed into pattern routing decisions based on A3C model. We mainly use L-shaped patterns to finish routing. Considering that there are different numbers of GCELLs in different routings and the features within GCELLs may too scattered to reflect the routing environments of the patterns, we calculate pattern features based on net segments. For an L-shaped pattern route, we take the inflection point as the demarcation, then two net segments can be obtained. Feature extraction is carried out by net segments, the details are listed below:

Net density value. We first construct a bounding box for each net and record the GCELLs that are involed in the bounding box. Then the number of bounding boxes that each GCELL associated with is obtained. Net density calculates the average number of bounding boxes within the current net segment. This feature provides the network relation of the net segment.

Congestion prediction value. This feature is computed based on the prediction results from the proposed GAT-based congestion predictor. We calculate the average values of future congestion of the current net segment using the congestion values predicted for each GCELL.

Capacity ratio value. We calculate the capacity of each GCELL based on the routing track setting from the def file. For a routing net, there are two GCELLs to be connected and four net segments from two L-shaped patterns. We set the net segments that pass through the same connected GCELL into the same category, and in turn divide the four net segments into two parts. For each part, we calculate the local capacity ratio between the two net segments to provide the capacity comparison features. The capacity of a net segment in this feature is computed as follow:

没看懂

$$CAP_{n\_m} = \frac{Capacity_{n\_m}}{\sum_{i=1}^{p} Capacity_{i\_m}}, \tag{1}$$

where n is the number of pattern, m is the class number of the net segments and p is the total number of all patterns. In the proposed method, n and m can be set to 1 or 2, p is 2. $Capacity_{n\_m}$ indicates the capacity of the m-th net segment of the n-th pattern. In order to keep the features consistent, we use $1 - CAP_{n\_m}$ as the capacity ratio value feature of the m-th segment of the n-th pattern.

C. DRL setup

Based on the above description, we define the state space, action space and reward function as follows:

State space. A state is the input data of the DRL model and it is a representation of features for the current net. Our state representation has 12 dimensions with three parts, including the capacity ratio value, the net density value and the congestion prediction value for each net segment in an L-shaped pattern, which have been discussed in the last section.

Action space. The action for our DRL model is to select one suitable pattern and perform routing.

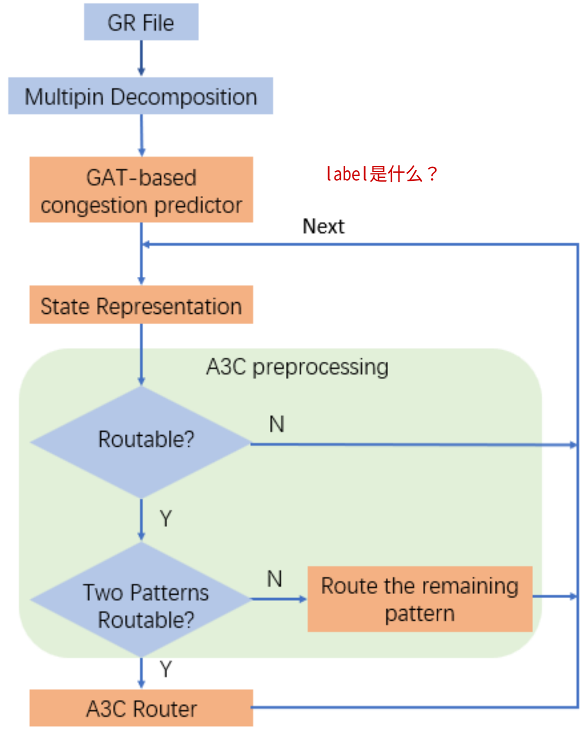
Reward function. After performing the selected action, the model will obtain feedback from the environment, which is provided by the reward function. Our reward function is defined as:

$$R = \begin{cases} 0, terminal = false, \\ -N_{fail}, terminal = true. \end{cases} \tag{2}$$

N_{fail} indicates the number of nets that failed. Each epoch when all nets of the layout have been performed, $terminal = true$. The reward function encourages the model to select the globally optimal routing pattern to reduce the number of failed nets.

D. Routing flow

Given a routing file, we first partition the layouts into GCELLs and calculate the capacity value for each GCELL according to the routing track setting. Then we decompose each multi-pin net into two-pin nets by Minimum Spanning Tree algorithm. The global routing problem is solved by the pattern routing method and the A3C model is introduced to decide the global optimal pattern. Before global routing, a GAT-based predictor is proposed to predict the future congestion for each GCELL. The predictor takes the extracted features as input for each GCELL and outputs the predicted congestion values. Based on the congestion result, the state representation is constructed as the input data of the A3C model. To enhance the routing efficiency, a preprocessing stage is designed for the A3C model, which is used for fast processing of unroutable



label是什么？

Fig. 3. The flow of the proposed method.

TABLE I
DETAILED INFORMATION OF THE CIRCUIT DESIGNS.

ISPD18 benchmark			
Design	**# Macros**	**# Nets**	**# two-pin Nets**
test1	0	3153	10964
test2	0	36834	112002
test3	4	36700	112391
test4	4	72410	240193
test5	8	72394	239914
test6	0	107701	359306
test7	16	179863	597451
test8	16	179863	597649
test9	0	178858	596639
test10	0	182000	613868

nets and nets that only one pattern routable. For nets with both patterns routable, the model will analyze the globally optimal decisions. The routing flow is presented in Figure 3.

V. EXPERIMENTAL RESULTS

In this paper, both the training and inference of the algorithm are performed on a machine with 3.60GHz CPU and one NVIDIA GeForce RTX 2080ti graphics card.

A. Experiment Setup

1) Parameter Settings: The proposed algorithm is implemented in python. For congestion predictor, the GAT network is implemented based on tf_geometric framework. The network structure of our GAT-based predictor consists of several parts, including two GAT layers from tf_geometric framework, a mean pool layer, and a fully connected layer. Gradient

这是什么，看看

TABLE II
THE RESULTS OF THE GAT-BASED CONGESTION PREDICTION.

% Accuracy	% Precision_C	% Precision_NC
76.08	82.41	75.10

descent optimizer is implemented with a learning rate of 0.0001. For A3C model, the number of workers for A3C model is 4 and the learning rate is set to be 0.001. The max episode is set to be 2000, while we usually stop the training when it converges. The time for finishing training is less than 15 hours.

2) Training and Test Set: We evaluate the proposed method on the ISPD18 benchmark from the ICCAD2019 global routing contest. Detailed information on each design is presented in Table I. We first train the GAT-based predictor and obtain the congestion features. Each design is routed by the Innovus tool only once to calculate the congestion labels and the layout is divided into GCELLs. When all data is prepared, we randomly choose 20% of the GCELLs with congestion and 20% of the GCELLs without congestion in each design for training set, and the remaining GCELLs for test set. For the A3C-based router, to obtain a well-trained model with efficient training, we randomly select ispd18_test5 as the training file and the other designs are the unseen test set.

B. Results

The exprimental result of the GAT-based congestion predictor is presented in Table II. In columns 1-3 are the accuracy, precision of the congested GCELLs, precision of the Non-congested GCELLs. The result shows that the GAT-based congestion predictor provides reliable features to help the A3C router determines patterns that may be less congested in the future. The final experimental results on ISPD18 global routing contest benchmark are presented in Table III. In columns 1-4 are the name of each design, the number of nets that fail to route, routability, and running time. From the tables we can observe that the proposed method achieves a total average routability of over 99.9% and the router has successfully routed 6 totally unseen designs without failure. Moreover, with more than 320,000 networks to be routed, the average number of failed nets in our approach is only around 6. The fail rate is less than 0.002%, which indicates the performance of the proposed algorithm compared with the routability of 96.8% in [12].

There are few studies of routing based on machine learning, therefore, in order to better analyze the performance of the proposed method, we implement another special version, OursI, which uses cost function to make decision for pattern routing. We use the average net density value, the average congestion prediction value and the average capacity ratio value of each pattern to construct the cost function in OursI and the pattern with less cost is selected to route. The comparison result is presented as Table IV, which indicates the performance of the proposed method.

TABLE III
ROUTING RESULTS OF THE PROPOSED MODEL ON PUBLIC BENCHMARK.

ISPD18 benchmark

Design	**# FailNet**	**% Routability**	**Time(s)**
test1	0	100.0	5.50
test2	0	100.0	48.70
test3	0	100.0	49.47
test4	0	100.0	116.46
test6	2	100.0	167.02
test7	25	99.9	295.19
test8	30	99.9	293.23
test9	0	100.0	269.23
test10	0	100.0	303.87
Average	6.33	99.99	172.09

TABLE IV
COMPARISON RESULT BETWEEN THE PROPOSED METHOD AND THE SPECIAL VERSION OURSI ON PUBLIC BENCHMARK.

这是什么！

ISPD18 benchmark

Design	**# FailNet**		**% Routability**	
	Ours	**OursI**	**Ours**	**OursI**
test1	0	0	100.0	100.0
test2	0	97	100.0	99.99
test3	0	118	100.0	99.9
test4	0	9	100.0	99.9
test6	2	23	100.0	100.0
test7	25	89	99.9	99.9
test8	30	80	99.9	99.9
test9	0	0	100.0	100.0
test10	0	0	100.0	100.0
Average	6.33	46.22	99.99	99.97

实验没有对比

VI. CONCLUSION

In this paper, we propose a congestion-aware reinforcement learning model for the global routing problem. A3C model is introduced to solve pattern routing. We propose a GAT-based congestion predictor to obtain global congestion features and a net segment mode for efficient feature extraction is introduced. The proposed method achieves an average routability of over 99.9% with the average failed net number around 6. In the future, we will further enhance the performance of the DRL global router.

REFERENCES

[1] Cunxi Yu and Zhiru Zhang, "Painting on PIacement: Forecasting Routing Congestion using Conditional Generative Adversarial Nets," 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2019, pp. 1-6.

[2] Tabrizi, Aysa Fakheri, Logan M. Rakai, Nima Karimpour Darav, Ismail Bustany, Laleh Behjat, Shuchang Xu and Andrew A. Kennings. "A Machine Learning Framework to Identify Detailed Routing Short Violations from a Placed Netlist." 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC) (2018): 1-6.

[3] Xie, Zhiyao, Yu-Hung Huang, Guan-Qi Fang, Haoxing Ren, Shao-Yun Fang, Yiran Chen and Nvidia Corporation. "RouteNet: Routability prediction for Mixed-Size Designs Using Convolutional Neural Network." 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (2018): 1-8.

[4] Liang, Rongjian, Hua Xiang, Diwesh Pandey, Lakshmi N. Reddy, Shyam Ramji, Gi-Joon Nam and Jiang Hu. "DRC Hotspot Prediction at Sub-10nm Process Nodes Using Customized Convolutional Network." 2020 International Symposium on Physical Design (2020).

[5] Hung, Wei-Tse, Jun Huang, Yih-Chih Chou, Cheng-Hong Tsai and Mango Chia-Tso Chao. "Transforming Global Routing Report into DRC Violation Map with Convolutional Neural Network." 2020 International Symposium on Physical Design (2020).

[6] Yu, Tao-Chun, Shao-Yun Fang, Hsien-Shih Chiu, Kai-Shun Hu, Philip Hui-Yuh Tai, Cindy Chin-Fang Shen and Henry Sheng. "Pin Accessibility Prediction and Optimization with Deep Learning-based Pin Pattern Recognition*." 2019 56th ACM/IEEE Design Automation Conference (DAC) (2019): 1-6.

[7] Yu, Tao-Chun, Shao-Yun Fang, Hsien-Shih Chiu, Kai-Shun Hu, Philip Hui-Yuh Tai, Cindy Chin-Fang Shen and Henry Sheng. "Lookahead Placement Optimization with Cell Library-based Pin Accessibility Prediction via Active Learning." 2020 International Symposium on Physical Design (2020).

[8] Mirhoseini, Azalia, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim M. Songhori, Shen Wang, Young-Joon Lee, et al. "Chip Placement with Deep Reinforcement Learning." ArXiv abs/2004.10746 (2020).

[9] Jain, Sambhav R. and Kye Okabe. "Training a Fully Convolutional Neural Network to Route Integrated Circuits." ArXiv abs/1706.08948 (2017).

[10] Liao, Haiguang, Wentai Zhang, Xuliang Dong, Barnabás Póczos, Kenji Shimada and Levent Burak Kara. "A Deep Reinforcement Learning Approach for Global Routing." ArXiv abs/1906.08809 (2019).

[11] Gandhi, Upma, Ismail Bustany, William Swartz and Laleh Behjat. "A Reinforcement Learning-Based Framework for Solving Physical Design Routing Problem in the Absence of Large Test Sets." 2019 ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD) (2019): 1-6.

[12] Utyamishev, Dmitry and Inna Partin-Vaisband. "Late Breaking Results: A Neural Network that Routes ICs." 2020 57th ACM/IEEE Design Automation Conference (DAC) (2020): 1-2.

[13] Kastner, Ryan, Elaheh Bozorgzadeh and Majid Sarrafzadeh. "Pattern routing: use and theory for increasing predictability andavoiding coupling." IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 21 (2002): 777-790.

[14] Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning." ArXiv abs/1602.01783 (2016).

[15] Ghose, Amur, Vincent Zhang, Yingxue Zhang, Dong Li, Wulong Liu and Mark J. Coates. "Generalizable Cross-Graph Embedding for GNN-based Congestion Prediction." 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) (2021): 1-9.

[16] Velickovic, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio and Yoshua Bengio. "Graph Attention Networks." ArXiv abs/1710.10903 (2018).

[17] Chanpuriya, Sudhanshu and Cameron Musco. "InfiniteWalk: Deep Network Embeddings as Laplacian Embeddings with a Nonlinearity." The 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (2020).