ISSN PRINT 2319 1775 Online 2320 7876

Research paper© 2012 IJFANS. All Rights Reserved, **UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022**

EXPLORE GRAPH NEURAL ARCHITECTURE SEARCH BASED ON DISTINCT SEARCH ALGORITHMS

Dr. Kanay Barik
Ghatsila College, Ghatsila

Abstract

Introduction - Graph Neural Architecture Search (GNAS) is an intriguing field that combines graph theory and neural network architecture search. There are various algorithms designed for GNAS, each with its own approach to discovering optimal graph neural network architectures.

Objective - The main intention of the research is to determine a distinct search algorithm under GNAS which assists to identify the defined search space. The research examines the benefits and drawbacks of each search algorithm approach, gives thorough explanations of each strategy, and offers the required comparisons.

Methodology - In order to effectively fulfil the study purpose and research questions, the article used non-experimental research under the quantitative research technique. The focus of the non-experimental study is on gathering pertinent information from big databases to improve our understanding of GNNs, GNAs, Search Algorithm and other recent advances in the field.

Results - The study found that Graph-NAS has gained prominence as a research area because of its ability to get around certain challenges that occur with manually creating GNN models. The comparative study of the existing algorithm illustrates that every design has certain loopholes and challenges. Thus, further study is essential to explore the outcome of these algorithms from practical grounds.

Keywords : Graph Neural Architecture Search, GNN, Search algorithm, Neural Architecture Search.

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

1. Introduction

Automating the process of finding the best architectures for GNN is the main goal of GNAS, a subfield of NAS. Neural networks that are specifically engineered to cope with graph-structured data are known as GNNs, and they are especially helpful for tasks like graph categorization, link prediction, and node classification (Zoph & Le, 2016). The goal of GNAS is to reduce the amount of manual work required to create GNN architectures, enabling the automatic identification of models that perform well in a variety of graph-based practices (Gao, Y et, al., 2021). The efficiency and effectiveness of GNAS are greatly impacted by the search algorithms used, the architecture of the search space, and the evaluation plan.

Search Algorithm, where the search method specifies how each potential GNN model in the search space will be sorted until the best model is identified. It is important to take computing space and calculation time into account while constructing the search algorithm (Zhou, K., et al., 2022). An algorithm is considered to be superior if it can arrive at the best answer more quickly and requires less computational power. Finding a trade-off between outstanding efficiency, affordable cost, and enhanced efficiency using the search algorithm is therefore the primary challenge (Li, W et al., 2020). Numerous search algorithms that are now in use have been used, such as differentiable search, evolution learning, RS, RL, and BO algorithm.

A supervised neural network that works well for applications that are focused on graphs and nodes is called a graph neural network. It is a GNN extension that keeps the features of the random walk and recursive models (Nunes & Pappa, 2020). The GNN is a more advanced form of recursive neural network since it can calculate any kind of graph, including cyclic, directed, and undirected ones. It can also handle targeted applications without the need for an initial processing stage (Chen, Z. et al., 2020). GNNs are very effective in applications like molecular chemistry, social network analysis, and recommendation systems that use graph-structured data.

Despite several characteristic graphs, neural networks possess several challenges, such as space and time complexity issues, to handle specific components like the “architecture component, attention function, aggregation function, activation function, and hyperparameters”, which are responsible for their dropout and learning rate (Guan, C. et al., 2022). In order to identify an effective graph neural network for a specific task, it requires investigation or testing of several graph neural architectures before choosing the best option, and the procedure is time taken and nearly impossible for large-scale graft data. Subsequently, several scholars emphasise automatically identifying the most suitable graph neural network for a particular task, encouraging a nobel concept in the academic world which is called “graph neural architecture search”. In this research, the paper is trying to explore several search algorithms that assist in

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

identifying the most suitable graph for the network from the search space and provide an exclusive outlook of the graph-NAS design.

Fig. 1 provides an overview of the NAS structure (Ren, P et al., 2021). According to the search space these function sets produce, NAS often starts with a collection of predetermined function sets and employs a controller to get a large number of potential neural designs. Subsequently, the potential architectures undergo training on the training dataset and are subsequently graded relayed on their reliability on the validation set. A collection of additional potential neural architectures may then be found by modifying the search method based on the order of data associated with the candidate neural design. The process of finding the optimal neural structure is stopped when the criterion for termination is met. After that, the test set's performance is assessed using the selected neural architecture.

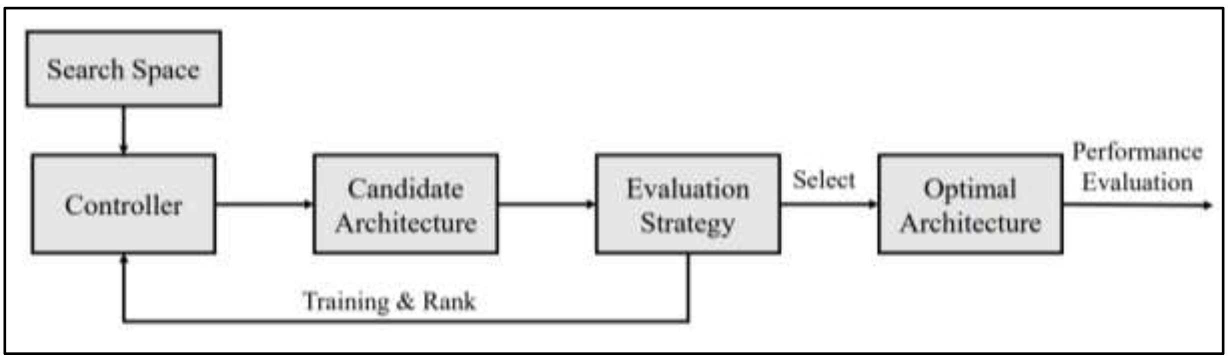

Figure 1: Overview of NAS (Ren, P et al., 2021).

2. Literature Reviews

GNAS has emerged as a potent technique for automatically identifying GNN designs that are appropriate for various applications. But because the present efficiency assessment algorithms in GNAS are operationally costly for giant networks and facing issues like durability collapse problems, existing methodologies are unable to handle large-scale graphs. The study proposes (Guan, C et al., 2022), “Graph Architecture Search at Scale (GAUSS)”, addresses these issues by creating an effective lightweight supernet and utilising combined architecture-graph sampling to handle large-scale graphs. Specifically, a single-path, one-shot supernet based on graph sampling is suggested to lessen the computational load. Their approach to resolving the problem of consistency collapse concerns involves explicitly evaluating combined architecture-graph sampling using an architectural significance sampling technique and an innovative collaborative learning technique on the sampled sub-graphs.

Another study (Qin, Y., et al., 2022) highlights two major obstacles substantially impeding the further study of GraphNAS. First, there are sometimes unfair comparisons made across research

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

publications since there is no agreement on the scientific setup, making the experimental findings inconsistent and even unreproducible. Second, GraphNAS is very inefficient and unavailable to investigators without the ability to perform large-scale computation since it frequently requires complex computations. The paper suggests NAS-Bench-Graph, a customised benchmark that enables uniform, fast, and repeatable assessments for GraphNAS, as a solution to these problems. The paper suggests an evaluation process based on principle evaluation. Fair, completely repeatable, and effective comparisons are made possible by the look-up table that provides the performance of GNN architectures based on our suggested benchmark, requiring no further processing.

The promise of neural architecture search (NAS) in identifying the best designs for language and visual modelling task learning has been demonstrated. Nevertheless, there are two reasons why the current NAS techniques cannot be effectively used for the GNN search issue (Zhou, K., et al., 2022). Firstly, learning sensitive performance changes with small architectural alterations in GNNs is not possible with the large-step discovery in the typical controller. Secondly, the multitude of GNNs that make up the search space make it impossible for them to directly embrace parameter sharing to accelerate the search process. An automated graph neural network (AGNN) methodology is proposed to address these issues by rapidly determining the best GNN design. In particular, a small-step exploration of the architectural space is intended by a strengthened conservative controller. In order to automate the degree of weight transfer across GNNs and expedite the validation process, a unique restricted parameter sharing technique is given. It saves computing time and prevents the need for initial training. When compared to both conventional search techniques and currently available human-invented models, experimental findings on the predefined database show that the design found by AGNN provides the highest level of efficiency and search effectiveness.

GNN are widely employed for the analysis of non-Euclidean data, including biological and social network data. Graph neural network design is a labour-intensive process that necessitates extensive domain expertise, notwithstanding its effectiveness. In this research (Gao, Y et al., 2019) paper offers an autonomous technique for searching the optimal design of NAS relay on reinforcement learning, which is called “Graph Neural Architecture Search (GraphNAS)”. In particular, GraphNAS trains a recursive paradigm that is expert with reinforcement learning to optimise the anticipated preciseness of the produced frameworks on an experimental data set, after which it employs a recursive paradigm in order to produce variable-length phrases instead which represent the structures of graph neural networks. In regard to testing set efficiency, GraphNAS can create a unique network structure for node classification tasks that is on par with the most accurate human-invented design.

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

The major problem associated with ML algorithms is to identify patterns and make generalised predictions on these data. Meanwhile this problem has been effectively solved using GNN. They gained popularity following the notion of convolution's extension to the graph realm. Nevertheless, they have a lot of hyper-parameters, and right now, algorithms or experiential perception are used in their hand-crafted design and optimization. NAS techniques seem to offer a promising answer to this issue. This research (Nunes & Pappa, 2020) takes this path by comparing two NAS approaches to GNN optimization: an evolutionary algorithm-based approach and a reinforcement learning-based approach. Findings take into account seven datasets spanning two search spaces and demonstrate that both approaches achieve accuracy comparable to a random search.

2.1 Research Gap

GNN and Graph-NAS are trendy topics in the mathematical world and attract the attention of several scientific scholars. Several of these existing studies were either exploring specific algorithms and their associations with graph-NAs, while others explored search space perspectives. None of the research papers emphasise exploring the search algorithms associated with Graph NAS. Henceforth, this research provides an overview of existing search algorithms related to graph-NAS, along with their limitations and advantages.

2.2 Research Question

1. Explore a distinct type of search algorithm applied in GNAS?
2. Determine the advantages and limitations of these search algorithms?

2.3 Importance of the Study

The significance of the paper is it provides a thorough worldwide review of the current Graph-NAS frameworks, including the issues that are currently acknowledged in the field of graph structure. The research examines the benefits and drawbacks of each search algorithm approach, gives thorough explanations of each strategy, and offers the required comparisons. The study also explores over a number of Graph-NAS search algorithms, assesses their drawbacks, and suggests interesting lines of inquiry for future work that balance scalability and efficiency.

2.4 Research Objectives

The main intention of the research is to determine a distinct search algorithm under Graph Neural Architecture Search which assists to identify the defined search space. Moreover, These algorithms guide the search process by iteratively proposing and evaluating different GNN architectures.

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

2.5 Scope and Limitation

The research's focus is on investigating GNAs from the standpoint of its search algorithm. Random Search (RS), Reinforcement Learning (RL), Evolution Learning (EL), Bayesian Optimization (BO) method, and Differentiable Search are the five main search algorithms that GNAs now use. While the paper focuses solely on investigating the search for graph neural architecture using different search algorithms, it neglects various other factors that are necessary to identify the best architectures for graph neural networks, including search space, which establishes a collection of GNN models, and performance evaluation, which determines a GNN model's quality. This acts as the limitation for the study and open source for future research.

3. Research Methodology

The study is non-experimental in nature since it heavily relies on secondary investigation, negating the necessity for any kind of experimentation. The focus of the non-experimental study is on gathering pertinent information from big databases to improve our understanding of GNNs, GNAs, Search Algorithm and other recent advances in the field. Non-experimental research often falls into the descriptive or theoretical category and uses suitable methods to describe a phenomenon and investigate the relationship between two or more variables that are the focus of the investigation.

3.1 Research Approach

This study uses a quantitative research strategy to investigate search algorithms in graph neural architecture search. In order to effectively fulfil the study purpose and research questionnaire, the article used non-experimental research under the quantitative research technique.

In this study, the longitudinal research methodology is appropriately adopted. According to longitudinal research, the most concentric and appropriate methodology for the research perspective as it can effectively and adequately fulfil the study's objective, which is to explore Graph neural architecture search based on distinct search algorithms.

4. Analysis of Study

Question 1 : Explore a distinct type of search algorithm applied in GNAS?

This study is mainly focused on distant types of search and gordhan such as Reinforcement learning, Evolution learning, Differentiable search, Random search, and Bayesian optimization. These algorithms explore the approach which is best suitable for obtaining effective solutions from the search space. It emphasises on quality and the accuracy of the network architecture

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

proposed. These search approaches can be appropriately utilised to generate effective solutions of GNN architecture.

- **Reinforcement Learning**

In the machine learning paradigm known as reinforcement learning (RL), an agent picks up decision choice capacity through connection with the neighbouring environment. The agent's goal is to develop a policy—a strategy—that maximises the cumulative reward over time. It gets feedback in the form of rewards or punishments (Zoph & Le, 2016) In RL-based Graph-NAS techniques, the search space reflects the surroundings, the management system, which is a neural network designed to produce an optimal design (referred to as a submodel) based on the search space over a period of time, represents the agent, and the assessment of the produced model's efficiency represents the reward. Recurrent networks are essentially used as supervisors by RL-based architectures to create explanations of GNN models (Gao, Y. et al., 2019). The rewards of the algorithms are then computed as feedback to optimise the anticipated efficacy of the resulting GNN systems.

- **Bayesian Optimization (BO)**

BO approach utilises a probabilistic function with an initial probability to calculate subsequent probabilities. Since forecasting and estimates of uncertainties are revised in response to fresh observations, the whole procedure is adaptive. BO consists of two primary parts: an acquisition function that determines the next sampling location and a Bayesian statistical model, also known as a surrogate function (SF), which models the objective function (Zhang, W et al., 2022). A probability distribution is constructed over trained data which is already used by the SF in BO, and the “acquisition operation” is utilised to effectively look for candidate instances before choosing them to assess the actual “objective function”. At every stage, the “acquisition function” is tuned to identify the ideal sample for the subsequent assessment. After then, the procedure is repeated until convergence using an updated version of the model. Only some scholars (Ma, L. et al., 2019; Zhou, H et al., 2019) have employed BO to solve the Graph-NAS problem, despite BO's better performance in NAS, and their outcomes are not encouraging. In particular, BO is nearly hard to execute successfully for large-scale data and computationally costly.

- **Evolution Learning**

The general population-based metaheuristic optimization technique known as "evolution learning" is based on selection by chance and natural genetic principles. There are several variations of this method; the genetic algorithm, also called the GA, is the one that is most frequently used in NAS frameworks (Nunes & Pappa, 2020). The population, often known as

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022**

persons, is the most significant part of the GA. A person is made up of genes or solution components. In the iterative process of genetic algorithm generation (GA), an initialised population's members are evaluated based on a fitness function, and the best person or individuals from the previous generation are used to create a new population (Liu, Y et al., 2021). The population in the Graph-NAS issue is created from the SS, and the genes represent a GNN framework description. Moreover, due to the vast breadth of the SS and the time-consuming nature of training a single GNN, particularly with a big graph dataset, it is nearly impossible to determine the fitness of every individual (Qin, Y et al., 2022).

- **Differentiable Search**

The fundamental concept of the differentiable search approach is integrating training and architectural sampling into a supernet in order to minimise resource overhead. This concept suggests creating a continuous search space as opposed to looking through a specific collection of possibilities (Zhao, H et al., 2020). Building an interconnected system (supernet) by stacking merged processes is the fundamental concept behind the differentiable searches for Graph-NAS. It applies all of the candidates' actions to each layer for each mix operation, then we blend their results linearly. The purpose of the amalgamated coefficients is to serve as chosen variables. After that, the researcher (Li, W et al., 2020; Elsken, T et al., 2019) conducts an architectural search using the minimization of a loss operation, like cross-entropy loss. The operation that yields the biggest coefficient in every layer is then selected to form the final architecture. Compared to earlier search algorithms, the parameterization approach used by the continuous relaxation scheme is less noisy and simpler to train (Wei, L et al., 2023). Additionally, this search technique can perform faster and with higher search quality than earlier search algorithms.

- **Random Search**

In RS, random submodels are created from the search space. Several researchers have tested RS. It employed a controlled RS to assess the significance of a particular option in relation to other alternatives of a comparable component in order to lessen the risk associated with this method. The method chooses the other components at random and tests a specified amount of GNN algorithms with the option to be assessed (Nunes & Pappa, 2020). The method evaluates the ranking distribution and assigns a performance ranking to the chosen GNN models for each option. The pick that matters the most is chosen for the ultimate GNNs (Ren, P et al., 2021).

Question 2: Determine the advantages and limitations of these search algorithms?

While the majority of the previously mentioned research methodologies have demonstrated outstanding effectiveness, it is important to acknowledge that each one has certain downsides.

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

The goal of the article is to compare the aforementioned search tactics while highlighting their benefits and limitations (Ren, P et al., 2021; Li, W et al., 2020; Elsken, T et al., 2019).

Reinforcement Learning (RL)

Advantages-

- Adjusts during the search process based on performance feedback.
- It has the ability to find intricate patterns inside the search space.

Limitations-

- RL algorithms might not use data efficiently. It is possible to repurpose previous experiences by using methods such as experience replay.
- The computational cost of graph-NAS can be high, particularly when dealing with large search areas.
- Managing exploration-exploitation trade-offs is crucial.

Random Search

Advantages-

- Straightforward and simple to use.
- Can occasionally find efficient designs with low processing demands.

Limitations-

- Ineffective when it comes to exploring.
- It could take a lot of experiments to identify the best designs.

Evolutionary Learning

Advantages-

- Analogous to the principle of natural selection, it investigates a wide variety of architectural designs.
- Ideally suited for computationally effective parallelization as well.

Limitations-

- May lead to less-than-ideal outcomes.
- Meticulous adjustment of evolutionary factors is necessary.

Bayesian Optimization (BO) Algorithm

Advantages-

- Effectively looks through the search space.
- Adjusts over time in light of previous assessments.

Limitations-

- Needs a surrogate model to be defined.

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

- Computationally costly at times.

Differentiable Search

Advantages-

- Makes effective use of differentiability in optimization.
- Can be easily combined with optimization techniques based on gradients.

Limitations-

- Restricted to search spaces that can differ.
- Possibly ineffective in handling discrete or irregular spaces.

The comparative study of all these algorithm can be illustrated in the table below-

Algorithm	Based on Principle	Benefit	Challenges
Random Search	Analyse the performance of graph neural network topologies by selecting samples at random.	Simple, convenient and sometimes surprisingly generates effective architectures.	Inefficient and may require a large number of trials to find optimal architectures.
Bayesian Optimization	Use Bayesian optimization to direct the search and model the performance of graph neural network topologies as a probabilistic function.	Effectively investigates the search space, changing over time in response to previous analyses.	Computationally costly.
Genetic Algorithms	Employ evolutionary operators like crossover, mutation, and selection to gradually develop a population of graph neural network topologies.	Examines a wide variety of designs and emulates the process of natural selection.	May lead to computationally costly and inefficient solutions.
Reinforcement	View the architectural	Adjusts based on	RL model training

ISSN PRINT 2319 1775 Online 2320 7876

 UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

Learning	search procedure as a series of decision-making tasks and employ reinforcement learning to acquire a policy for producing architectures.	performance comments obtained throughout the search.	might need a lot of resources, and trade-offs between exploration and exploitation must be carefully considered.
Evolutionary Algorithms	Graph-based evolutionary algorithms are utilised to evolve graph neural network topologies, which should be represented as graphs.	Ideally matched to neural networks' graph-structured architecture.	With graph structure and evolution, complexity increases.

Table 1: Comparative study of distinct search algorithms.

5. Results

The study found that Graph-NAS has gained prominence as a research area because of its ability to get around certain challenges that occur with manually creating GNN models. While graph-NAS frameworks have demonstrated their potential to generate excellent GNN models, a reliable trade-off between superior performance, affordable price, and scalability need to be discovered, since current approaches are unable to sustain low costs for high adaptability while maintaining high performance.

The most suitable solution for a model cannot be found by a search algorithm outside of the solutions contained in the search space. Thus, all state-of-the-art GNN model architectures should be taken into account in an adequate search space. The comparative study of the existing algorithm is illustrated that every design has certain loopholes and challenges. Thus, further study is essential to explore the outcome of these algorithms from practical grounds.

A promising area of future study might be to standardise the search space with just influential functions while taking into account all of the best model designs, given the variety of the best GNN model architectures. Reducing the linearity between Graph-NAS processing time and data size might be accomplished by researching how parallel computing can be used to current search techniques.

ISSN PRINT 2319 1775 Online 2320 7876

6. Conclusion

GNAS may be used with a variety of algorithms, including random search, genetic algorithms, Bayesian optimization, reinforcement learning, and meta-learning. These algorithms propose and assess several GNN topologies iteratively, guiding the search process. Each method has advantages and disadvantages, and the best option is determined by a number of variables, including the desired amount of automation, computational resources, and issue complexity. Hybrid techniques are frequently used in research, integrating many algorithms to take use of their complementing advantages can be considered in future research. Moreover, the subject of graph-based architectural search (GNA) is a fast-moving target, with research on novel approaches to effective and efficient neural architecture search continuing.

References

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. The Journal of Machine Learning Research, 20(1), 1997-2017.

Gao, Y., Yang, H., Zhang, P., Zhou, C., & Hu, Y. (2019). Graphnas: Graph neural architecture search with reinforcement learning. arXiv preprint arXiv:1904.09981.

Gao, Y., Yang, H., Zhang, P., Zhou, C., & Hu, Y. (2021, January). Graph neural architecture search. In International joint conference on artificial intelligence. International Joint Conference on Artificial Intelligence.

Guan, C., Wang, X., Chen, H., Zhang, Z., & Zhu, W. (2022, June). Large-scale graph neural architecture search. In the International Conference on Machine Learning (pp. 7968-7981). PMLR.

Li, W., Gong, S., & Zhu, X. (2020, April). Neural graph embedding for neural architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 4707-4714).

Liu, Y., Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Tan, K. C. (2021). A survey on evolutionary neural architecture search. IEEE transactions on neural networks and learning systems.

Ma, L., Cui, J., & Yang, B. (2019, October). Deep neural architecture search with deep graph bayesian optimization. In IEEE/WIC/ACM International Conference on Web Intelligence (pp. 500-507).

ISSN PRINT 2319 1775 Online 2320 7876

Research paper© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 11A, 2022

Nunes, M., & Pappa, G. L. (2020). Neural architecture search in graph neural networks. In Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9 (pp. 302-317). Springer International Publishing.

Pham, H., Guan, M., Zoph, B., Le, Q., & Dean, J. (2018, July). Efficient neural architecture search via parameters sharing. In the International conference on machine learning (pp. 4095-4104). PMLR.

Qin, Y., Zhang, Z., Wang, X., Zhang, Z., & Zhu, W. (2022). NAS-Bench-Graph: Benchmarking graph neural architecture search. Advances in Neural Information Processing Systems, 35, 54-69.

Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Chen, X., & Wang, X. (2021). A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR), 54(4), 1-34.

Wistuba, M., Rawat, A., & Pedapati, T. (2019). A survey on neural architecture search. arXiv preprint arXiv:1905.01392.

Zhang, W., Lin, Z., Shen, Y., Li, Y., Yang, Z., & Cui, B. (2022, June). Deep and Flexible Graph Neural Architecture Search. In the International Conference on Machine Learning (pp. 26362-26374). PMLR.

Zhao, H., Wei, L., & Yao, Q. (2020). Simplifying architecture search for graph neural networks. arXiv preprint arXiv:2008.11652.

Zhou, H., Yang, M., Wang, J., & Pan, W. (2019, May). Bayesnas: A bayesian approach for neural architecture search. In International conference on machine learning (pp. 7603-7613). PMLR.

Zhou, K., Huang, X., Song, Q., Chen, R., & Hu, X. (2022). Auto-gnn: Neural architecture search of graph neural networks. Frontiers in Big Data, 5, 1029307.

Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.