Enhancing Reinforcement Learning through Graph Neural Networks: A Novel Approach

[1]Priya Singh, [2]Dr. Md. Abdul Aziz Al Aman, [3]Dr. Saroj Kumar

[1]Department of Computer Science Techno India University Kolkata, West Bengal

[2]Department of CSE-AI Techno India University Kolkata, West Bengal

[3]AiLabs-Artificial Intelligence, DGC DataCore Systems (India) pvt. Ltd. Kolkata, West Bengal

DOI: https://doi.org/10.51244/IJRSI.2025.1210000046

Received: 10 October 2025; Accepted: 14 October 2025; Published: 01 November 2025

ABSTRACT

Reinforcement Learning (RL) has showcased remarkable success in various domains. However, its performance often degrades in the environment with complex structures and distributed rewards. Graph-Based Reinforcement Learning (GBRL) is an approach that combines the strengths of Graph Theory with Reinforcement Learning to optimize complex decision making problems in any networked system. This paper proposes an approach of integrating Reinforcement Learning approaches with Graph Neural Networks (GNNs)to enhance the learning pipeline and model structured data by utilising their capacity. We present an approach that uses GNNs represented as graphs that enables RL agents to get dependencies between entities and access information through them. This paper exhibits GBRL techniques and their application in different domains. A framework of GBRL methods and its advantages over RL methods in working on graph-based data. This work highlights the synergy between graph-based learning and decision-making, offering a promising direction for solving high-dimensional and structured RL tasks more effectively. We also summarize the key challenges and the open research directions in this field.

Keywords– Graph-Based Reinforcement Learning (GBRL), Reinforcement Learning (RL), Graph Neural Networks (GNN), Convolution Neural Networks (CNNs), Deep Reinforcement Learning (DRL), Temporal Difference (TD), Asynchronous Advantage Actor-Critic (A3C), Advantage Actor-Critic (A2C), Deterministic Policy Gradient (DPG), Deep Deterministic Policy Gradient (DDPG), Multi-Relational Graph (MR-Graph), Multi-Relational GNN, Markov's Decision Process (MDP), Monte Carlo (MC) methods, Deep Q-Network (DQN)

INTRODUCTION

Recently the development of networked systems has boosted the interest in the methods that can efficiently handle complex and structured data which these systems generate. GBRL emerges as an encouraging method that utilises the power of reinforcement learning with graph theory to address decision making problems. The relationships within entities represented as graphs, GBRL allows refine understanding of problem space, enabling an effective learning and decision making. Conventional RL approaches struggle with interconnection and complexity present in graph structured data. GBRL deals with these challenges by using graph methods with RL algorithms, giving better exploitation and exploration of the network structure. This has tremendously implicated various domains, like biological systems, social networks, communication networks, and transportation systems, where optimization and understanding between interactions among entities is crucial. This paper aims to give a broad overview of GBRL, exploring its approaches, potential future directions and applications. The paper begins with the fundamental concepts of RL and graph theory, followed by the

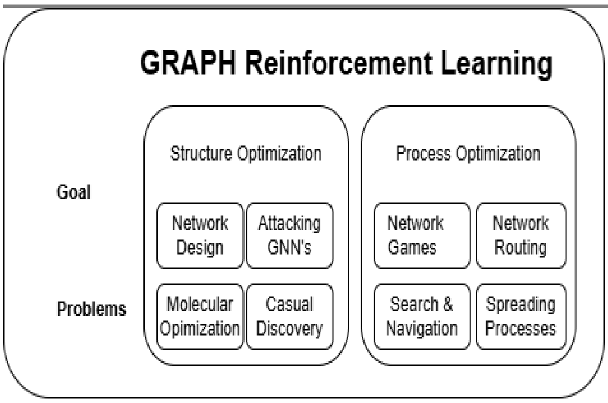

Figure 1**:** Graph Based Reinforcement Learning (GBRL) : Goals with problems that GBRL solves details of GBRL techniques. With this paper, we intend to highlight advantages of GBRL and identify key challenges with the proposed strategies for advancing this field.

A. Graph Based Reinforcement Learning

The GBRL techniques pertinence are broad, as they exhibit versatility of the graph frameworks formed of interconnected entities with their relationships. Beyond the description of the problems, it is normal to address questions on optimization, where the goal is intervening the system to improve properties. The requirements for GBRL can be summarized as:

1. Graphs are appropriate for problems under consideration, with edges and nodes having clear semantics.

2. Decision-making is possible to intervene in systems beyond mere observation.

3. To solve a problem approximately is acceptable.

LITERATURE REVIEW

The progress of GNNs enables effective representation learning in various fields [1] including social networks, NLP [2, 3], recommender system [4], social events [5, 6, 7], computer vision and physics [8]. GNN models can show optimal performance over huge datasets on different tasks like node clustering [9, 10], link prediction [11, 12, 13], graph classification [14, 15, 16, 17] etc. As per the difference in modeling for actual graph data, we can divide GNN methods into homogeneous GNN, heterogeneous GNN, and multiple graph learning models. Homogeneous GNN are those methods that overlook data type of nodes or attributes of edges on graphs. Classical methods include GCN [18], Graph-SAGE [19] and GAT [20]. Heterogeneous GNN approaches consider the heterogeneity of node or edge types. Considering diverse edges in actual data, more relational GNN including R-GCN [21], FdGars [22], SemiGNN [23] and GraphConsis [24] were developed. Apart from homogeneous and heterogeneous GNNs, multi-graph neural network models [25, 26, 27, 28, 29] fuse the multiple characterizations to learn the embedding of graph data. Recently, GNNs have been used to compute and model graph-based data to predict relationships in graphs and improve the ability to reason. These models establish rules for graph-based data, from biological network analysis to social media, optimization and network modeling [30]. GNN emerges as a tool to solve graph mining problems. With the advancement of technology, RL algorithms have generated many directions of development. The value based basic algorithms are Q-Learning algorithms [31] and DQN [32] algorithms, that use value functions to gauge and reduce occurrence of the optimal situations. whereas policy-based algorithms directly perform iterative calculations on the policy such as PPO [33]. The Actor-Critic RL methods combine advantages of both value-based and policy-based. Formerly there were a few attempts to integrate GNNs and RL. A model, DGN+GNN [34] is used to generalise network topologies, where GNNs allow the RL agents to operate on different networks. A RL based graph model, G2S+BERT+RL [35] is for natural question generation, GNN is made to process the directed graph passage. Other works [36, 37, 38] investigate how GNNs improve the generalisation ability of RL. Also there are numerous studies that show RL can be optimized on graphs. DeepPath [39], GraphNAS [40], Policy-GNN [41], RL-HGNN [42] are a few examples.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

METHODOLOGIES AND APPROACHES

For this subsection, the goal is the introduction of the RL algorithms and to draw their connections to graphs. Graph RL methods presented rely on a huge variety of algorithms, each of which is characterized by distinct principles and assumptions. It is rather not possible to cover all algorithms, it is necessary to give some of them for solving MDPs.

A. Dimensions of RL Algorithms

There are different RL methodologies used in different applications.

Model-based algorithm and Model-free algorithm: To be specific, let's assume state space (S) and action space (A) known as a model M = (P, R) refers to having an estimate of transition function (P) and reward functions (R). Model-based algorithms can add knowledge to greatly optimize learning speed up. It can be used in the form of mathematical set descriptions that define P and R fully. It is beneficial where authenticity is expensive to generate and where there is a negative impact on executing poor policies (e.g., robotics). When equipped with the perspective of the world, policy is planned by the agent, either decision-time planning, or background planning. In Model-free algorithms, P is corresponding to the density estimation problem, while R is the Supervised Learning problem. It has simple learning architectures. This has higher model complexity, i.e. they need more interactions with the environment to train.

On-policy and off-policy: The differentiation relies on the occurrence of these two policies: behavior policy which is used to interact with the environment and target policy is the policy that is being learned. The behavior and target policy are same in on-policy methods while they are different in off-policy algorithms. On-policy methods are used as special cases while off-policy methods are more flexible.

Sample-based and Temporal Difference: The sample-based methods rely upon environmental interaction samples, rather than complete knowledge of MDP. Temporal Difference (TD) methods are based on samples of experience. They are biased with less variance.

RL Methods

Recently, RL methods have achieved a huge success in varied applications that automatically handles sequential decision based problems in environments via decision making and goal directed learning, these types of methods achieve huge success in many games. RL is developed as an MDP, a sequential-decision making mathematical model whose actions affect current rewards, the subsequent states with future rewards. In MDP, prediction problems and control problems are implemented using dynamic programming.

The definition of MDP says the tuple $\{S, A, T, R, p(s_0), \gamma\}$, where S is set of all possible states, which are generalization of environment, A means set of actions which are adopted in states, which are agent's all possible actions, $R : S\times A\times S\rightarrow R$ shows reward function, which are rewards returned by environment to agent after an action's execution, $T : S\times A\rightarrow p(S)$ is denoted by state transition function, $\gamma \in [0, 1]$ indicates the discount factor, that is treated as hyperparameter of agent and uses to promote faster availability of rewards to agent. Figure 2 shows the process of agent's interaction with the environment.

Here, to find a policy π is our goal that maximises Q(s, a) the expected action-value function, (1) defines the target policy.

$$\pi^* = \underset{\Pi}{argmax} Q(s,a)$$

$$= \underset{\Pi}{argmax} E_{n,T} \left[\sum_{k=0}^{K} \gamma^k r_{t+k} | s_t = s, a_t = a\right] \qquad (1)$$

where π^* is equal or better than all other policies. Further we provide fundamental concepts of the different popular RL methods.

a

INTERNATIONAL JOURNAL OF RESEARCH AND SCIENTIFIC INNOVATION (IJRSI)
ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

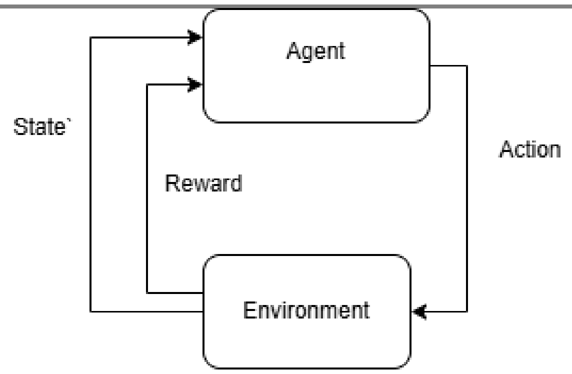

Figure 2. Process of interaction between the agent and the environment.

1) Q-Learning: Q-learning and Temporal Difference (TD) learning methods are off-policy learners, which are an important result to early research on RL. The values on the Q-table are directly updated by target policy in Q-learning to achieve selection of optimal policy, whereas behavior policy employs ε-greedy policy for the semi-random exploration of environment. The action-value function Q has a learning goal to be learned in Q-learning. By direct approximation the optimal action-value function q∗ are learned, so that Q-learning algorithm can formulate as (2).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2)$$

where the equation (2) denotes state s_t which explores an environment with behavior policy that is based on values in Q-table at timestamp *t*. It performs an action *a*, the reward as *R* and a new state as s_{t+1} is obtained based on environmental feedback. The given equation has been executed to update the latest Q-table. It will continue updation of new state s_{t+1}, after completion of the operation until termination time *t*.

2) Reinforce: REINFORCE doesn't optimize directly based on policy space, rather learns parameterized policy without being known about the intermediate value estimation function, that uses Monte Carlo method to learn policy parameters with estimated returns and full trace. This method makes a neural-network based policy which takes inputs as states and gives output as probability distribution in operation space. This policy π parameterize θ a set of weights so that $\pi(s;\ \theta) \equiv \pi(s)$, that is action probability distribution on state, and REINFORCE is updated (3).

$$\Delta\omega_{i,j} = \alpha_{i,j}\,(r - b_{i,j})\,\frac{\vartheta}{\vartheta\omega_{i,j}} \ln g_i \quad (3)$$

where a non-negative learning factor is $\alpha_{i,j}$, discounted reward value is denoted by *r*, and representation function of state $b_{i,j}$ is for reducing variance of gradient estimate.

3) Actor-Critic: Actor-Critic algorithm uses a value function and a parameterized policy, where a better $\hat{A}(s, a)$ estimate of computation of the policy gradient is provided by the value function. This algorithm elaborates both state and policy value functions by combining advantages of policy gradient and value function based algorithms. The Actor in this method, denotes policy function for learning policy that can be obtained as any number of rewards as possible. Whereas the Critic represents an estimated value function that is used for evaluating estimated value of current policy. Figure 3 here showcases the framework of the Actor-Critic algorithm. For the RL algorithms, the basic framework like A3C, A2C, DPG, DDPG is the Actor-Critic Architecture.

4) Deep Q-Network: Q-learning algorithm along with Q-table in large scale graph-structured data suffers from the problem of large number of intermediate state values, causing dimensional disaster. To address this challenge, a method is proposed which combines neural networks and value-function approximation through action or state space reduction and function approximation. DQN is misused to learn the policies by the leveraging of deep neural networks. It is proposed to use Q-learning algorithms for representing states with graph

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

embeddings, according to properties of graph-structured data, to minimize impact of non-Euclidean structure on Q-table scale. In general, DQN combines Q-learning with deep learning and approximates action-value function with deep neural-networks,

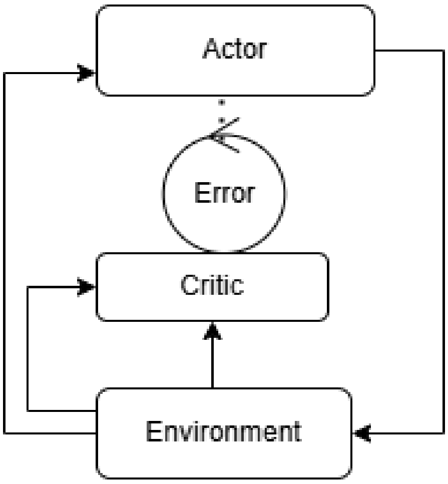

Figure 3. Framework for Actor-Critic algorithm. Here from the given environment, the Actor receives a state and selects the action to be performed. In the meantime, Critic receives its current state and state that is generated by previous interactions and calculates TD error to update Actor and Critic.

and obtains $\{s_t, a_t, r_t, s_{t+1}\}$ the trace from interaction of agent with environment. It can implement successful policies from high dimensional sensory inputs by using the end-to-end RL. The loss function $L(\theta_t)$ of this method is given by (4).

$$L(\theta_t) = E[(r + \gamma \max_{a+1} Q_{\theta_{t-1}}(s_{t+1}, a_{t+1}) - Q_{\theta_t}(s_t, a_t))^2] \quad (4)$$

To stabilize the training process DQN gives two techniques: (1) A buffer replay to reuse past experiences. (2) A separate target-network that can be periodically updated.

B. Graph Embedding Methods

Graph Neural Networks term means a deep embedding method. Let's recall the graph *G = (V, E)* in which v_i nodes have feature vectors $x_{v_i}$and, optionally, with $x_{e_{i.j}}$as edge features. The goal here is to get h_{v_i} an embedding vector for each of the nodes that captures features along with structure of interactions on a given graph. The calculation or generation of the embedding vectors takes place in $l \in 1, 2, ..., L$ layers, where L represents the final layer. $h_{v_i}^{(l)}$can be used to denote the embedding of v_i nodes in the l layer. Notation $W^{(l)}$, indexed by the subscript, represents a weight matrix which depicts the block of learnable parameters in the layer l of the Graph Neural Network model. Unless the embeddings were initialized with node features, $h_{v_i}^{(0)} = x_{v_i}, \forall v_i \in V$.

1) Message Passing Neural Network: Message Passing Neural Network (MPNN) (Gilmer et al., 2017) is the framework which abstracts many of the graph learning architectures, along with serving as one of the useful conceptual models present for deep embedding methods. It has layers that apply *M(l)* a message function and *U (l) a* vertex-update function, to compute embeddings.

$$m_{v_i}^{(l+1)} = \sum_{v_i \in N(v_i)} M^{(l)}(h_{v_i}^{(l)}, h_{v_j}^{(l)}, x_{e_{i,j}})$$

$$h_{v_i}^{(l+1)} = U^{(l)}(h_{v_i}^{(l)}, m_{v_i}^{(l+1)}) \quad (5)$$

a

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue X October 2025

where $\mathcal{N}(v_i)$ represents the open neighborhood of node v_i . Afterwards, *I* the readout function is applied to calculate

embedding for entire graph from the final node embeddings set: $I(\{h_{v_i}^{(L)}|\})I(\{h\ (L)vi|vi \in V\})$. The message function and vertex-update functions are learned. The readout function can either be fixed a priori or learned.

2)Graph Convolutional Network: Graph Convolutional Network method (GCN) (Kipf & Welling, 2017) is simpler to merely rely on multiplication of features of nodes with weight matrix together with degree-based normalization. It is a first-order approximation of local spectral filters on the graphs. As it can be created as the series of matrix multiplication, it scales well to the large graphs with millions of edges while giving superior performance to the other methods of embedding at the time. It can be formulated as:

$$H_{v_i}^{(l+1)} = ReLU(W_1^{(l)} \sum_{v_j \in \mathcal{N}[v_i]} \frac{h_{v_j}^{(l)}}{\sqrt{(1+deg(v_i)(1+deg(v_j))}}) \qquad (6)$$

where degree of node v_i is indicated by *deg(v_i)*, and closed neighborhood of node v_i by $\mathcal{N}[v_i]$, including all its neighbors as well as v_i itself.

3) Graph Attention Network: Graph Attention Network model (GAT) (Veličković et al., 2018) proposed use of the attention mechanisms (Bahdanau et al., 2016), it is considered as a way to do flexible aggregation on the neighbor features. Coefficients of learnable aggregation enables an increase in the expressibility of the model, which translates gains in the predictive performances over GCN for the classification of nodes. Let $\zeta_{i,j}^{(l)}$ be denoted as the attention coefficient that shows importance of features of node v_j to the features of node v_i in the layer l. It is computed as:

$$\zeta_{i,j}^{(l)} = \frac{exp(\ LeakyReLU(\theta^T[W_1^{(l)} h_{v_i}^{(l)} || W_1^{(l)} h_{v_j}^{(l)} || W_2^{(l)} x_{e_{i,j}}]))}{\sum_{v_k \in \mathcal{N}[v_i]} exp(LeakyReLU(\theta^T[W_1^{(l)} h_{v_i}^{(l)} || W_1^{(l)} h_k^{(l)} || W_2^{(l)} x_{e_{i,k}}))} \qquad (7)$$

Where the exponential function is represented by *exp(x)* = e^x, weight vector is θ that parametrize attention mechanism and concatenation is denoted by [·‖·]. The activation function LeakyReLU(x), which gives the non-zero values for negative inputs, according to α_{LR} the small slope, is equal to $\alpha_{LR}x$ if x < 0, and x otherwise. For the given attention coefficient, node embeddings can be calculated according to the rule given below.

$$h_{v_i}^{(l+1)} = \sum_{v_j \in \mathcal{N}[v_i]} \zeta_{i,j}^{(l)} W_1^{(l)} h_{v_j}^{(l)} \qquad (8)$$

Connections Between GNN and RL

This section will discuss the relationship between GNN and RL in context to GBRL framework. Learning techniques that are designed for the operations on graphs are used commonly as the function approximators which are considered as part of RL algorithms. GBRL methods are goal driven and constructive, which allows for the flexibility in finding embeddings relevant to the objective function that is to be optimized without any fine-grained supervision signal. Alternatively, primitive graph learning benchmark to rely on the supervised learning and availability of granular examples. The work on molecular optimization is an exception (You et al., 2018a), which has used a GCN as a differentiator trained on the example molecules which provide a part of the reward signal. It is noteworthy as another line of recent work that connects GNNs and RL. In this work, a representation of a graph is constructed where nodes are states, and edges represent transitions to be determined by actions.

CHALLENGES AND GAPS

Irrespective of the potential of GBRL, several gaps and challenges in this research are needed to be addressed. These challenges highlight the opportunities for further improvement of GBRL methods:

a

1) Scalability: Scalability to large-scaled graph-structured environments is a challenge of GBRL algorithms. Processing huge amounts of data efficiently and managing its computational complexity remains its significant hurdles.

2) Learning in Dynamic Environments: Many graphs are dynamic in the real world, with edges and nodes evolving over time. Recent GBRL methods struggle in adapting these changes, demanding algorithms capable of handling dynamic graphs for the development.

3) Generalization and Transfer Learning: Essuring GBRL models to generalize within different domains and tasks is crucial. Difficulties faced by current approaches in transferring knowledge to a new and unseen environment, with spotlighted need to improve transfer learning techniques.

4) Explainability and Interpretability: With increasingly complex GBRL models ensuring their decisions are interpretable and explainable for practical applications. Remains a significant challenge is enhancing transparency.

5) Quality and Data Sparsity: For effective learning high-grade with broad scope of graph data is essential. However, noise, data sparsity or missing information can inhibit performance of GBRL models. It is critical to develop methods for handling these data issues.

6) Multi-Agent Coordination: In multiple agents involved scenarios, ensuring an effective collaboration in a graph-structured environment and coordinating the actions of these agents can be challenging.

7) Evaluation and Benchmarking: For GBRL, the deficiency of standard criteria and evaluation metrics makes it difficult comparing different approaches and measuring progress. Making comprehensive criteria frameworks is necessary for the development of the field.

8) Societal and Ethical Implications: As implementation of GBRL techniques to real-world problems, it is crucial to understand and address their societal and ethical implications. Ensuring accountability, fairness and reducing biases in GBRL models are the most important considerations.

With these challenges and gaps addressed, the GBRL techniques can progress in developing more practical, scalable and robust solutions for complex decision-making problems.

FUTURE RESEARCH SCOPE

GBRL has great capabilities for addressing different kinds of complex real-world problems. Whereas, there are several open questions and challenges that need further investigation. The below given areas represent various directions for future research in the field of GBRL:

1) *Efficiency and Scalability:* It is crucial to develop algorithms which can further scale to large and complex graph-structured environments. Optimization of computational resources and improvement of scalability of GBRL methods should be future research focus.

2) *Generalization and Transfer Learning:* An important area to explore is enhancement of capabilities of GBRL algorithms in transferring learned knowledge within different domains along with the generalisation of new and unseen environments. Investigation techniques for new domain adaptations and transfer learning will also help in achieving this goal.

3) *Multiple Agent Systems:* GBRL extension to multiple agent systems, where agents learn and interact simultaneously within these graph-structured environments, presents a unique challenge. Further research in these directions can lead to advancement in coordination among agents and collaborative decision-making.

4) *Dynamic Graphs:* As many real-world graphs are dynamic in nature, with edges and nodes changing over time. Development of GBRL methods that can adapt such dynamic graphs while learning from data is a thrilling research direction.

5) Interpretability and Explainability: For practical applications, ensuring the transparency of GBRL models and their decision-making processes is essential. Future work must focus on enhancing the explainability of the GBRL algorithms making them much more trustworthy and more accessible to users.

6) Real-World Applications: Further research in the emerging fields such as cybersecurity, healthcare, smart cities etc is needed to explore the potential of GBRL.

CONCLUSION

GBRL represents a remarkable advance in the machine learning domain, offering distinct capabilities for handling complicated problems of decision-making in graph environments. By combining the concepts of RL with graph theory, GBRL enables efficient exploitation and exploration of relationships within a network or networks, enhancing performances across applications. This paper highlights the principles, methodologies and various applications of GBRL. It also identifies the key challenges and gaps in current research, emphasising on the need for adaptive, scalable and interpretable GBRLs. The way to more robust and practical implementation of GBRL can be addressed through these challenges in real-world scenarios. In conclusion, GBRL provides a framework to better understand and optimize complex networked systems. As the research progresses in this area, we predict that GBRL plays a pivotal role in the addressing of some pressing challenges in this interconnected world.

REFERENCES

1. Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. Machine learning 8, 3-4 (1992), 279–292.
2. David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bom barell, Timothy Hirzel, Al´an Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In Proceedings of the NIPS. 2224–2232.
3. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. nature 518, 7540 (2015), 529–533.
4. William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 1025–1035.
5. Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. Proceedings of the ICLR.
6. Guixiang Ma, Lifang He, Chun-Ta Lu, Weixiang Shao, Philip S. Yu, Alex D Leow, and Ann B Ragin. 2017. Multi-view clustering with graph embedding for connectome analysis. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 127–136.
7. John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).
8. Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In Proceedings of the EMNLP. ACL, 564–573.
9. Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80). PMLR, Stockholmsm¨assan, Stockholm Sweden, 1587–1596.
10. Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In Proceedings of the NIPS. 4284–4295.
11. Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. 2018. Drug similarity integration through attentive multi-view graph auto-encoders. In Proceedings of the 27th International Joint Conference on Artificial Intelligence. 3477–3483.
12. Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. In Proceedings of the IJCAI. AAAI Press, 2609–2615.

13. Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In Proceedings of the WWW. 1063–1072.
14. Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In Proceedings of the ESWC. Springer, 593–607.
15. Petar Veliˇckovi´c, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. Pro ceedings of the ICLR.
16. Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In Proceedings of the NIPS. 4800–4810.
17. Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In Proceedings of the NIPS. 5165–5175.
18. Xi Zhang, Lifang He, Kun Chen, Yuan Luo, Jiayu Zhou, and Fei Wang. 2018. Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson’s disease. In AMIA Annual Symposium Proceedings, Vol. 2018. 1147.
19. Paul Almasan, Jos´e Su´arez-Varela, Arnau Badia-Sampera, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2019. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. arXiv preprint arXiv:1910.07421 (2019).
20. Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Genera tion. In Proceedings of the ICLR.
21. Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. 2019. Graphnas: Graph neural architecture search with reinforcement learning. arXiv preprint arXiv:1904.09981 (2019).
22. Di Jin, Ziyang Liu, Weihao Li, Dongxiao He, and Weixiong Zhang. 2019. Graph convolutional networks meet markov random fields: Semi supervised community detection in attribute networks. In Proceedings of the AAAI. AAAI Press, 152–159.
23. Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. 2019. GCN GAN: A non-linear temporal link prediction model for weighted dynamic networks. In Proceedings of the IEEE INFOCOM. 388–396.
24. Hao Peng, Jianxin Li, Qiran Gong, Yangqiu Song, Yuanxing Ning, Kunfeng Lai, and Philip S. Yu. 2019. Fine-grained event categorization with heterogeneous graph convolutional networks. In Proceedings of the IJCAI. AAAI Press, 3238–3245.
25. Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A Semi supervised Graph Attentive Network for Financial Fraud Detection. In Proceedings of the IEEE ICDM. 598–607.
26. Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. 2019. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In Proceedings of the World Wide Web Conference. 310–316.
27. Victor Bapst, Thomas Keck, A Grabska-Barwi´nska, Craig Donner, Ekin Dogus Cubuk, Samuel S Schoenholz, Annette Obika, Alexander WR Nelson, Trevor Back, Demis Hassabis, et al. 2020. Unveiling the predictive power of static structure in glassy systems.
28. Patrick Hart and Alois Knoll. 2020. Graph Neural Networks and Rein forcement Learning for Behavior Generation in Semantic Environments.
29. Jarom´ır Janisch, Tom´aˇs Pevny, and Viliam Lis ‘ y. 2020. Symbolic Relational Deep Reinforcement Learning based on Graph Neural Networks. arXiv preprint arXiv:2009.12462 (2020).
30. Kwei-Herng Lai, Daochen Zha, Kaixiong Zhou, and Xia Hu. 2020. Policy-GNN: Aggregation Optimization for Graph Neural Networks. In Proceedings of the ACM SIGKDD. New York, NY, USA, 461–471. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks • 39:43
31. Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. Proceedings of the SIGIR, 1569–1572.
32. Hao Peng, Jianxin Li, Qiran Gong, Yuanxin Ning, Senzhang Wang, and Lifang He. 2020. Motif-Matching Based Subgraph-Level Attentional Convolutional Network for Graph Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. 5387–5394.

33. Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. 2020. Exploiting Cross-session Information for Session-based Recommendation with Graph Neural Networks. ACM Transactions on Information Systems (TOIS) 38, 3 (2020), 1–23.
34. Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. IEEE Transactions on Knowledge and Data Engineering (2020).
35. Penghao Sun, Julong Lan, Junfei Li, Zehua Guo, and Yuxiang Hu. 2020. Combining deep reinforcement learning with graph neural networks for optimal VNF placement. IEEE Communications Letters 25, 1 (2020), 176–180.
36. Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems (2020).
37. Zhiqiang Zhong, Cheng-Te Li, and Jun Pang. 2020. Reinforcement Learning Enhanced Heterogeneous Graph Neural Network. arXiv preprint arXiv:2010.13735 (2020).
38. Yuwei Cao, Hao Peng, Jia Wu, Yingtong Dou, Jianxin Li, and Philip S. Yu. 2021. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. In Proceedings of the Web Conference 2021. Association for Computing Machinery, 3383–3395.
39. [39] Hao Peng, Jianxin Li, Yangqiu Song, Renyu Yang, Ranjan Rajiv, Philip S. Yu, and He Lifang. 2021. Streaming Social Event Detection and Evolution Discovery in Heterogeneous Information Networks. ACM Transactions on Knowledge Discovery from Data 15, 5 (2021).
40. Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip S. Yu, and Lifang He. 2021. Hierarchical taxonomy-aware and attentional graph capsule RCNNs for large-scale multi-label text classification. IEEE Transactions on Knowledge and Data Engineering 33, 6 (2021), 2505–2519.
41. Qingyun Sun, Hao Peng, Jianxin Li, Jia Wu, Yuanxing Ning, Phillip S. Yu, and Lifang He. 2021. SUGAR: Subgraph Neural Network with Reinforcement Pooling and Self-Supervised Mutual Information Mechanism. In Proceedings of the Web Conference. 2081–2091.
42. Yang Wang. 2021. Survey on Deep Multi-Modal Data Analytics: Collaboration, Rivalry, and Fusion. ACM Transactions on Multimedia Computing, Communications, and Applications 17, 1s, Article 10 (2021), 25 pages. In IEEE Intelligent Vehicles Symposium (IV). IEEE, 1589–1594