Constraint-Aware Probabilistic Packet Forwarding Based on Deep Reinforcement Learning

Guocheng Lin, Yang Xiao, and Jun Liu
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
Email: {linguocheng, zackxy, liujun}@bupt.edu.cn

***Abstract*—The growing demand for network resources has made routing optimization a critical challenge in ensuring quality of service requirements. Many researchers have introduced probability into forwarding to achieve finer control and have employed deep reinforcement learning (DRL) to enable autonomous control. However, current DRL-based probabilistic packet forwarding approaches are limited by the lack of constraints for forwarding and cross-comparisons with other routing approaches, which results in severe routing loops and a lack of effectiveness and persuasiveness. To address these issues, this paper proposes a constraint-aware probabilistic packet forwarding approach, which employs a novel joint graph attention network actor-critic structure and is trained using proximal policy optimization. Besides, this paper proposes a novel probabilistic packet forwarding protocol for forwarding constraints to ensure routing safety. To validate the effectiveness of the proposed approach, comprehensive experiments are conducted within the proposed network simulation framework based on knowledge-defined networking. The results across diverse topologies and loads show that the proposed approach significantly improves network quality of service and robustness compared to two state-of-the-art DRL-based probabilistic packet forwarding approaches, three state-of-the-art DRL-based routing approaches, and two conventional routing approaches.**

***Index Terms*—Probabilistic packet forwarding, deep reinforcement learning, routing optimization, quality of service.**

I. Introduction

The exponential surge in network demand has exerted unprecedented pressure on network infrastructures. In this case, internet service providers must manage diverse traffic patterns to ensure efficient and reliable routing. Consequently, research on routing optimization has become crucial for maintaining stable network performance. In past practices, conventional routing approaches like open shortest path first (OSPF) [1] and equal-cost multi-path (ECMP) [2] are straightforward and robust. However, these conventional routing approaches often fail to fully exploit the varied link resources among nodes, frequently leading to performance bottlenecks, congestion, and resource wastage under heavy traffic. Currently, a significant body of research has been developed around these conventional routing approaches [3]. The primary finding identified in these studies is that introducing randomness in packet forwarding for more precise control could significantly enhance routing performance.

Probabilistic packet forwarding is an approach that introduces randomness by forwarding packets in a data stream to multiple links with probabilities. Therefore, probabilistic packet forwarding can fully leverage link resources and effectively mitigate network congestion [4]. Numerous researchers have incorporated probabilistic packet forwarding into routing optimization. By dynamically maintaining a flow table with next-hop selection probabilities, the proposed method of Wang et al. [5] substantially enhances network load balancing while reducing average end-to-end (E2E) delay and packet loss rate. Mori et al. [6] proposed a probabilistic packet forwarding approach based on round-trip time, effectively lessening network congestion and content retrieval delays. However, these conventional probabilistic packet forwarding approaches struggle to fully comprehend the intricacies of evolving networks and autonomously adjust forwarding probabilities. Therefore, researchers have turned to augmenting the network awareness and autonomous optimization capabilities of probabilistic packet forwarding. Fortunately, advances in artificial intelligence offer novel avenues for these enhancements.

In recent years, deep reinforcement learning (DRL) has emerged as a promising solution in routing optimization due to its ability to extract high-dimensional features and manage complex system control tasks [7]. Increasingly, researchers are exploring the integration of DRL with probabilistic packet forwarding and have achieved remarkable results. Li et al. [8] proposed a graph attention proximal policy optimization algorithm, which enhances the robustness of routing to network changes by modifying the probability of selecting the next hop in routers. Chen et al. [9] designed a DRL-based probabilistic packet forwarding algorithm with the front-convergence actor-critic network that significantly reduces the delay and packet loss rate in delay-sensitive networks. Wang et al. [10] implemented an efficient adaptive probabilistic cognitive routing approach, achieving excellent performance in real-world networks. Zhou et al. [11] focused on link features and developed a probabilistic routing algorithm that can be reused in different topologies without retraining. Li et al. [12] introduced contrastive learning and graph neural networks into probabilistic packet forwarding, proposing a contrastive graph transformation routing algorithm to perceive unseen link failures without retraining.

However, despite the remarkable achievements of current DRL-based probabilistic packet forwarding approaches in au-

This work was supported by the National Natural Science Foundation of China under the Grant No. 62371057. *(Corresponding author: Yang Xiao.)*

ISBN 978-3-903176-72-0 © 2025 IFIP

tonomous controlling and network QoS performance, several critical issues remain unresolved as follows:

- Most existing studies on DRL-based probabilistic packet forwarding primarily focus on enhancing the performance of DRL algorithms, often neglecting key considerations related to probabilistic packet forwarding. As a result, the process of probabilistic packet forwarding in existing studies frequently lacks constraints awareness, leading to severe routing loops and substantial resource waste in networks.
- Existing DRL-based probabilistic packet forwarding approaches typically compare only with other probabilistic packet forwarding approaches rather than with alternative routing approaches. Furthermore, these studies often lack comprehensive experiments across various topology scales and load conditions, which may substantially undermine their persuasiveness and effectiveness.

To address the issues above, this paper proposes a constraint-aware probabilistic packet forwarding approach based on deep reinforcement learning, and extensive experiments have been conducted. The major contributions of this paper are as follows:

- We first propose an innovative probabilistic packet forwarding protocol designed to minimize routing loops and ensure routing safety. Subsequently, we propose a novel joint graph attention network-empowered proximal policy optimization (JGAT-PPO) algorithm, which effectively leverages topology and link information with the help of a novel joint graph attention network structure to enhance routing approach adaptability in dynamic networks and improve the QoS performance.
- We design and implement a packet-level network simulation framework based on the knowledge-defined networking architecture. This simulation can efficiently and reliably validate the effectiveness of the proposed JGAT-PPO and facilitate comparisons with not only probabilistic packet forwarding approaches but also different DRL-based and conventional routing approaches.
- We conduct extensive experiments under various network scales and traffic loads. We compare JGAT-PPO against two state-of-the-art DRL-based probabilistic packet forwarding approaches, three state-of-the-art DRL-based routing approaches, and two conventional ones. The experimental results demonstrate that JGAT-PPO outperforms the comparative approaches regarding QoS performance under different loads and topologies, minimizing the packet loss rate while reducing the average E2E delay as much as possible.

II. Probabilistic Packet Forwarding Protocol

A. Forwarding Approach

The proposed probabilistic packet forwarding protocol (PPFP) filters the next-hop candidate list for packets in each node by maintaining a topology table $\mathcal{D} = \{d_{ij}\}_{N\times N}$ and selects the next hop for packets from the candidate list by maintaining a probability table $\mathcal{P} = \{p_{ij}\}_{2\times E}$, as shown in **Algorithm 1**. Here, d_{ij} represents the shortest distance from node i to node j, p_{ij} represents the probability that node i selects its neighboring node j for packet forwarding, N represents the number of nodes, and E represents the number of links. The next hop is finally determined through both $\mathcal{P}$ and the roulette selection algorithm [13] in PPFP.

Algorithm 1: Probabilistic Packet Forwarding Protocol

```
Initialize the topology table $\mathcal{D} = \varnothing$ and the probability table $\mathcal{P} = \varnothing$;
Utilize breadth-first search to compute $\mathcal{D}$;
while current node $n_c$ receives a packet do
    Parse source node $n_s$ and destination node $n_d$;
    Initialize the next-hop candidate list $\mathcal{N} = \varnothing$;
    for each potential next hop $n_k$ at current node $n_c$ do
        Read $d_{n_k n_d}, d_{n_c n_d}, d_{n_k n_s}, d_{n_c n_s}$ from $\mathcal{D}$;
        if $(d_{n_k n_d} < d_{n_c n_d})$ or
              $(d_{n_k n_d} = d_{n_c n_d}$ and $d_{n_k n_s} > d_{n_c n_s})$ then
            $\mathcal{N} \leftarrow \mathcal{N} \cup \{n_k\}$;
        end
    end
    for each candidate node $n_k$ in $\mathcal{N}$ do
        $p_{n_c n_k} \leftarrow \frac{p_{n_c n_k}}{\sum_{n_j \in \mathcal{N}} p_{n_c n_j}}$;
    end
    Select next hop $n_n \in \mathcal{N}$ using calculated probabilities $\{p_{n_c n_k}\}_{n_k \in \mathcal{N}}$;
    Forward the packet to $n_n$;
end
```

B. Network Performance Metrics

The following network performance metrics are emphasized to demonstrate the effectiveness of the proposed approach.

1) Average End-to-End Delay: The end-to-end (E2E) delay d_k of a successfully arrived packet k is calculated as the sum of its transmission, propagation, queuing, and processing time. For the K packets that successfully reach their destinations at time step t, the average E2E delay $d(t)$ is defined as

$$d(t) = \frac{1}{K}\sum_{k=1}^{K} d_k. \tag{1}$$

2) Packet Loss Rate: The packet loss rate $lr(t)$ at time step t is defined as the ratio of the number of packets N_{loss} that fail to reach their destinations to the total number of transmitted packets N_{trans}:

$$lr(t) = \frac{N_{loss}}{N_{trans}}. \tag{2}$$

C. Optimization Objective

JGAT-PPO aims to minimize average E2E delay while ensuring a low packet loss rate. Accordingly, the optimization objective $o(t)$ at time step t is defined as

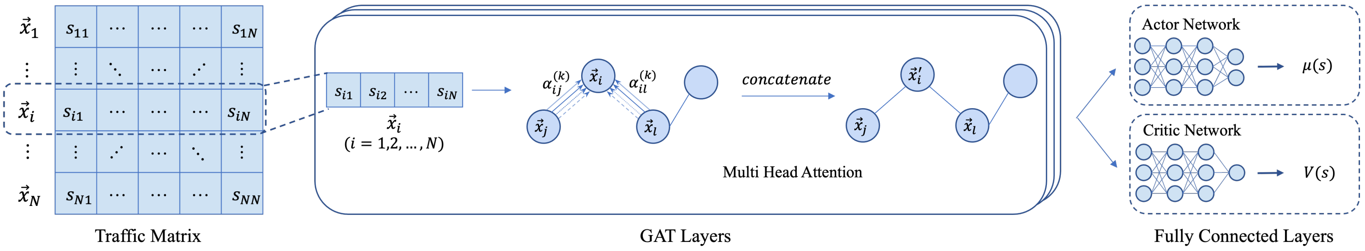


Fig. 1. The structure of the proposed JointGAT structure.

$$o(t) = \omega_1 \cdot U_\alpha(d(t)) + \omega_2 \cdot lr(t). \tag{3}$$

Here $\omega_1 + \omega_2 = 1$, where ω_1 and ω_2 represent the weights of average E2E delay and packet loss rate within the optimization objective, respectively. $U_\alpha(x) = \frac{x^{1-\alpha}}{1-\alpha}$ is a utility function designed to maintain average E2E delay and packet loss rate within a closely bounded range [9].

III. DRL-Based Constraint-Aware Probabilistic Packet Forwarding

A. Problem Formulation

The routing process can be modeled as a Markov decision process (MDP) to tackle the complexities of dynamic networks. The MDP is described by the tuple $\langle S, A, P, R \rangle$, comprising S as the state space, A as the action space, P as the state transition probabilities, and R as the reward function. The DRL agent continuously monitors the network state and performs actions, subsequently receiving rewards from the environment to optimize its policy. The definitions of the state space, action space, and reward function are as follows.

1) State Space: The state space is defined as the traffic matrix in the network:

$$S_t = \{s_{ij}^t\},\ i, j = 1, 2, ..., N, \tag{4}$$

where N represents the total number of nodes, and s_{ij}^t indicates the volume of traffic sent from node i to node j at time step t. For ease of differentiation, $s_{ij} = -1$ if there is no link between node i and node j.

2) Action Space: The action space is defined as the probability table $\mathcal{P}$ in the proposed PPFP, where the forwarding probabilities for each next hop at each node are recorded.

3) Reward Function: To prevent the DRL agent from excessively relying on immediate rewards and leading to policy oscillations, the cumulative reward function R_t at time step t is adopted and defined as

$$R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}, \tag{5}$$

where $\gamma \in (0, 1]$ is the discount factor that quantifies the influence of past rewards on the present, and r_t is the present reward at time step t. The goal of the DRL agent is to maximize R_t, while the optimization target is to minimize $o(t)$. Consequently, the present reward value is defined as $r_t = -o(t)$ to reduce the average E2E delay and packet loss rate within the network.

B. Joint Graph Attention Network Structure

Graph attention networks (GAT) [14] combine attentional mechanisms with graph neural networks (GNN) to effectively perceive and learn graph features. Li et al. [8] have demonstrated that GAT can effectively extract and utilize network features for routing optimization. In this section, we proposed a novel joint graph attention network structure (JointGAT), which combines GAT with the actor-critic structure to perceive network features and make informed decisions fully. As shown in Fig 1, both the actor and critic networks in JointGAT leverage the shared GAT layers and are updated jointly to enhance feature perception and accelerate learning.

JointGAT begins by initializing a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{F' \times F}$ and a learnable weight vector $\vec{a} \in \mathbb{R}^{2F'}$, where F' is the transformed feature dimension. Each time JointGAT receives the state input, it extracts node features $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, ...\vec{x}_N\} \in \mathbb{R}^F$ from the state. Each node feature vector $\vec{x}_n \in \mathbf{X}$ includes the outbound traffic intensities of node n. Subsequently, the node features are put into the shared GAT layers to extract graph features. At this time, a shared attention mechanism is employed for node i and each of its neighboring nodes j to compute the raw attention coefficients e_{ij} as

$$e_{ij} = \text{LeakyReLU}(\vec{a}^T[\mathbf{W}\vec{x}_i \| \mathbf{W}\vec{x}_j]). \tag{6}$$

Subsequently, the raw attention coefficients are normalized to obtain attention scores α_{ij} as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{n \in \mathcal{N}_i} \exp(e_{in})}, \tag{7}$$

where $\mathcal{N}_i$ represents the neighborhood set of node i. Finally, the attention scores α_{ij} are used to compute the new feature vector for node i. JointGAT utilizes a multi-head attention mechanism to enhance expressiveness, where the feature vectors from each attention head are aggregated through concatenation to form the extracted node feature as

$$\vec{x}'_i = \Big\|_{m=1}^{M} \sigma\Big(\sum\nolimits_{j \in \mathcal{N}_i} \alpha_{ij}^{(m)} \mathbf{W}^{(m)} \vec{x}_j\Big), \tag{8}$$

where M represents the number of attention heads, $\sigma(\cdot)$ represents the activation function, and $\alpha_{ij}^{(m)}$ and $\mathbf{W}^{(m)}$ respectively denote the attention coefficients and the learnable weight matrix for the m-th head.

Algorithm 2: JGAT-PPO

```
Initialize the JointGAT with random parameter $\theta$,
  learning rate $\eta$, and replay buffer $\mathcal{R} = \varnothing$;
Initialize the packet-level simulation framework $env$;
for $i$ = 1 to $N_{ep}$ do
    Reset $env$;
    for $j$ = 1 to $N_{step}$ do
        Observe current state $s_j$ from $env$;
        Construct the action distribution
          $\pi_\theta(a_j|s_j) = \mathcal{N}(\mu(s_j), \sigma^2\mathbf{I})$;
        Sample action $a_j \sim \pi_\theta(a_j|s_j)$ and record its
          probability $p_j = \pi_\theta(a_j|s_j)$;
        Take action $a_j$ in $env$ and obtain reward $r_j$;
        Store tuple $\langle s_j, a_j, p_j, r_j \rangle$ into $\mathcal{R}$;
    end
    Initialize a value buffer $\mathcal{V} = \varnothing$;
    for $k$ = 1 to $k_{update}$ do
        Obtain the state value $v_k$ by applying the critic
          network to $\mathcal{R}$ and store $v_k$ into $\mathcal{V}$;
        Calculate joint loss $\mathcal{L}_k$ using $\mathcal{R}$ and $\mathcal{V}$;
        Update the parameter $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_k$;
    end
    Clear $\mathcal{R}$ and $\mathcal{V}$;
    $\sigma \leftarrow \max(\sigma_{\min}, \sigma - \sigma_r)$,;
end
```

C. Clipped Proximal Policy Optimization with JointGAT

In this section, we proposed a novel joint graph attention network-empowered clipped proximal policy optimization algorithm (JGAT-PPO). JGAT-PPO adopts JointGAT as the actor-critic network structure and employs clipped proximal policy optimization (Clipped PPO) [15] for training. Clipped PPO is an on-policy DRL algorithm known for enhanced stability, improved sample efficiency, and superior generalization compared to other policy gradient algorithms. The loss function of Clipped PPO is defined as

$$L_t(\theta) = \mathbb{E}_t\Big\{ \min\Big[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}A(t), c_t(\theta)A(t)\Big]\Big\}, \tag{9}$$

$$A(t) = A(s_t|a_t) = R_t - V_t(s_t), \tag{10}$$

$$c_t(\theta) = \text{clip}(\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}, 1-\epsilon, 1+\epsilon), \tag{11}$$

where $\pi_\theta(a_t|s_t)$ represents the policy with parameter θ, and $c_t(\theta)$ is the clip function employed to limit the difference between the updated policy $\pi_{\theta'}(a_t|s_t)$ and the current policy $\pi_\theta(a_t|s_t)$. $A(t)$ is the advantage function, which calculates the difference between the cumulative reward R_t and the state value $V_t(s_t)$. $A(t)$ evaluates the benefit of executing action a_t relative to the expected cumulative reward.

During execution, JGAT-PPO employs a multivariate normal distribution to regulate exploration behavior. The mean value $\mu(s)$ of the distribution is output by the actor network, while the standard deviation σ of the distribution is controlled by the action decay mechanism to balance exploration and exploitation. At the beginning of training, the action decay mechanism sets the standard deviation of the distribution to the maximum action standard deviation $\sigma_{\max}$. With agent update, σ decays at a rate of σ_r until it reaches the minimum action standard deviation $\sigma_{\min}$. JGAT-PPO trains for N_{ep} episodes, each consisting of N_{step} steps. At the end of each episode, JGAT-PPO updates the parameters of JointGAT k_{update} times through the joint loss function $\mathcal{L}$ as

$$\mathcal{L} = \omega_A \cdot \mathcal{L}_A + \omega_C \cdot \mathcal{L}_C + \omega_E \cdot H(a), \tag{12}$$

$$\mathcal{L}_A = \frac{1}{T}\sum_{t=0}^{T} \min\Big[\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}A(t), c_t(\theta)A(t)\Big], \tag{13}$$

$$\mathcal{L}_C = \frac{1}{T}\sum_{t=0}^{T}\big[A(t)\big]^2, \tag{14}$$

where $\mathcal{L}_A$ and $\mathcal{L}_C$ represent the loss of the actor network and critic network, guiding the current policy towards the optimal policy. $H(a)$ is the entropy of the current action distribution, encouraging the agent to explore more policies and avoid premature convergence to sub-optimal policies. ω_A, ω_C, and ω_E are the weights for $\mathcal{L}_A$, $\mathcal{L}_C$, and $H(a)$, respectively. In summary, the pseudocode of JGAT-PPO is presented in **Algorithm 2**.

IV. Experiment Evaluation

A. Packet-Level Network Simulation Framework

Knowledge-defined networking (KDN) [16] is an emerging network architecture expected to play a crucial role in autonomous networks. KDN introduces intelligent decision-making and knowledge storage on top of software-defined networking [17] architecture, optimizing network behavior through the dynamic learning of network knowledge and intelligent inference. Inspired by KDN and RL4NET++ [18], we have designed and implemented a novel packet-level network simulation framework. This simulation framework integrates probabilistic packet forwarding and simulations of various routing methods, providing robust support for the validation of JGAT-PPO and comparisons with other state-of-the-art routing approaches.

The architecture of the proposed network simulation framework is illustrated in Fig. 2. Among these, the knowledge plane is the core component of the KDN architecture. The DRL module in the knowledge plane is implemented using Python and the PyTorch [19] framework, which transforms the state into knowledge and stores it in the knowledge storage. Then, the knowledge plane makes the action according to the knowledge storage. The action is then received by the INET [20] controller in the control plane, which converts the action into the routing policy and sends it to the data plane. In the proposed simulation framework, PPFP and other routing protocols are implemented in the INET controller. The data plane implemented by OMNeT++ [21] updates the

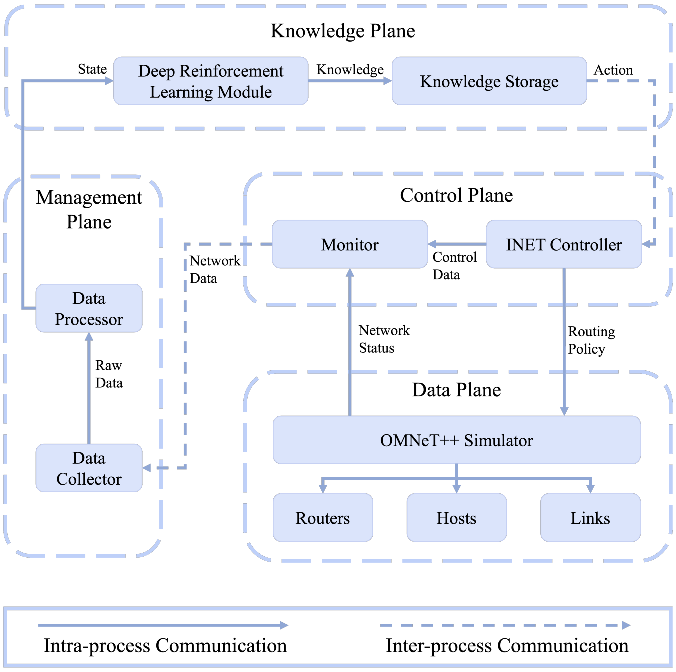


Fig. 2. The architecture of the proposed network simulation framework.

configuration of network devices based on the routing policies for packet forwarding and routing. During network operation, the monitor in the control plane periodically retrieves and organizes the network status and control data as network data for the management plane. The data collector in the management plane sends the raw data to the data processor. Ultimately, the data processor converts the raw data into the state and sends it to the DRL module in the knowledge plane. By repeating the abovementioned process, the proposed network simulation framework effectively simulates autonomous learning and intelligent decision-making of the network during operation.

The proposed network simulation framework employs different communication methods based on the characteristics of each plane to improve inter-plane communication efficiency, as illustrated in Fig. 2. Among these, intra-process communication is adopted within the same plane, between the data and control planes, and between the knowledge and management planes. Besides, ZeroMQ [22] is utilized to implement inter-process communication between the knowledge and control planes and between the control and management planes. With the help of these different communication methods, the proposed network simulation framework can efficiently retrieve network information for analysis.

B. Experiment Setup

The experimental platform is operated on Ubuntu 22.04 system with two Intel Xeon Gold 5218R CPUs and six Tesla T4 GPUs. The network simulation framework is implemented based on OMNeT++ 6.0.1, INET 4.5, and ZeroMQ 4.3.4. Besides, all DRL-based routing approaches are implemented using Python 3.11.4 and PyTorch 2.3.1. The learning parameters of JGAT-PPO is set as $\gamma = 0.9$, $\alpha = 0.6$, $\epsilon = 0.2$, $\omega_A = -1$, $\omega_C = 0.5$, $\omega_H = 0.1$, $\sigma_{\max} = 1.25$, $\sigma_{min} = 0.05$, and $\sigma_r = 0.01$.

JGAT-PPO aims to minimize the packet loss rate while maintaining a low average E2E delay. Therefore, the weights in the reward function are set to $\omega_1 = 0.4$ and $\omega_2 = 0.6$. The adaptive moment estimation optimizer (Adam) [23] is employed in JGAT-PPO with a learning rate of $1e-3$. The shared GAT layers in JGAT-PPO are configured with a hidden dimension of 64 and incorporate a four-head multi-attention mechanism. The number of neurons in the three fully connected layers of the actor network is 64, 32, and the value of action dimension, while the critic network consists of three fully connected layers with 64, 32, and 1 neurons. The output layer of the actor network adopts the Softmax function as the activation function, while the remaining layers of JointGAT adopt ReLU. The agent is trained over 200 episodes, each with 50 steps. After each episode, Monte Carlo estimation is employed 10 times to approximate rewards and refine the policy.

Three real network topologies of varying scales sourced from Topology Zoo [24] are adopted for experiments, including Gridnet (with 9 nodes and 20 edges), BtNorthAmerica (with 36 nodes and 76 edges), and Dfn (with 58 nodes and 87 edges) to enhance the generalizability of validation results. The topology diagrams for Gridnet, BtNorthAmerica, and Dfn are shown in Fig. 3. Due to computational resource constraints, a 1-second timeout is applied to packet-sending applications, considering a packet lost if it fails to reach the destination node within this time. Besides, the bandwidth of all links within the topologies is scaled to 1 Mbps, and the link transmission delay is set to 2 ms in the experiments. Each simulation time step lasts 1 second, during which nodes continuously send packets to all the other nodes to simulate a complex network environment. The packet-sending interval follows a Poisson distribution. The simulation framework provides statistics for the current time step to the agent only after confirming that all packets sent within the time step have either reached their destination nodes or been lost.

C. Evaluation Metrics

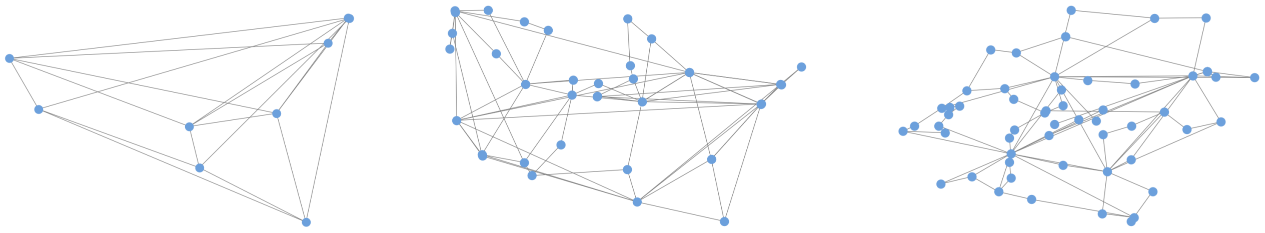
To facilitate better comparison, the load capacity of a link is estimated based on the degree of its adjacent nodes. For the link between node i and node j, the load capacity L_{ij} is defined as

$$L_{ij} = \frac{N-1}{d_i} + \frac{N-1}{d_j}, \tag{15}$$

where d_i and d_j are the degrees of node i and node j, respectively. N represents the total number of nodes in the network. Then, the maximum load capacity $L_{\max}$ of the network topology is defined as

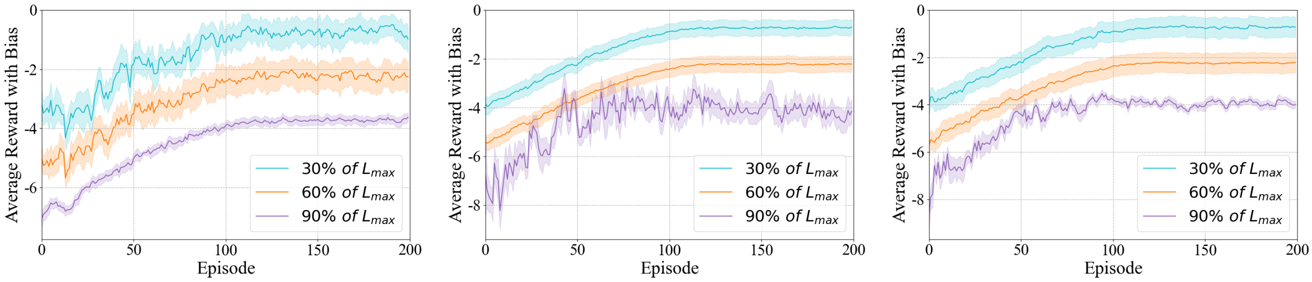
$$L_{\max} = \max_{(i,j)\in\mathcal{C}} \left(\frac{N-1}{d_i} + \frac{N-1}{d_j} \right), \tag{16}$$

where $\mathcal{C}$ is the set of network links. According to the settings mentioned in Section IV-B, each link in the network has the



(a) Gridnet (b) BtNorthAmerica (c) Dfn

Fig. 3. The topology diagrams for Gridnet, BtNorthAmerica, and Dfn.



(a) Convergence performance on Gridnet (b) Convergence performance on BtNorthAmerica (c) Convergence performance on Dfn

Fig. 4. Convergence performance on Gridnet, BtNorthAmerica, and Dfn.

same bandwidth B. Therefore, the flow rate R_f of each data stream under load L can be calculated as

$$R_f = \frac{B}{2L}. \tag{17}$$

In the experiments, traffic intensity under 30%, 60%, and 90% of the maximum load L_{max} on Gridnet, BtNorthAmerica, and Dfn is used to compare the performance of the proposed routing approach with other routing approaches under different load conditions.

D. Benchmark Approaches

For adequate validation, this paper evaluates the performance of JGAT-PPO compared to two state-of-the-art DRL-based probabilistic packet forwarding approaches, three state-of-the-art DRL-based routing approaches, and two conventional routing approaches, including:

- *Proximal Policy Optimization with Front-Convergent Actor-Critic Network (PPO-FCACN)* [9]: PPO-FCACN utilizes a novel front-convergent actor-critic network architecture for DRL-based probabilistic packet forwarding, effectively meeting the QoS requirements of delay-sensitive applications in networks.
- *Constrastive Graph Transformer Routing (CGTR)* [12]: CGTR introduces GNN and constrastive learning to DRL-based probabilistic packet forwarding, enhancing the robustness to unseen link failures in networks without retraining during routing.
- *Message Passing Deep Reinforcement Learning (MP-DRL)* [25]: MPDRL leverages GNN and DRL to extract knowledge from message passing between links, effectively achieving load balancing and improving network QoS performance.
- *Graph Neural Networks-Based Flexible Online Routing (G-Routing)* [26]: G-Routing utilizes GNN and DRL to predict network performance metrics and flexibly adjust forwarding objectives during routing, which exhibits excellent convergence speed and reliability under network changes.
- *Twin Delayed Deep Deterministic Policy Gradient in Mult-Agent System (TD3-MAS)* [27] TD3-MAS uses the twin delayed deep deterministic policy gradient algorithm to split traffic across multiple paths, significantly reducing network costs and ensuring maximum average E2E delay requirements.
- *Open Shortest Path First (OSPF)* [1]: OSPF directly forwards packets based on the shortest path from the source node to the destination node, aiming to improve routing efficiency and reduce transmission costs in networks.
- *Equal-Cost Multi-Path (ECMP)* [2]: ECMP evenly forwards packets in traffic to multiple equal-cost paths, effectively improving network tolerance to failures and resource utilization of links.

E. Simulation Results

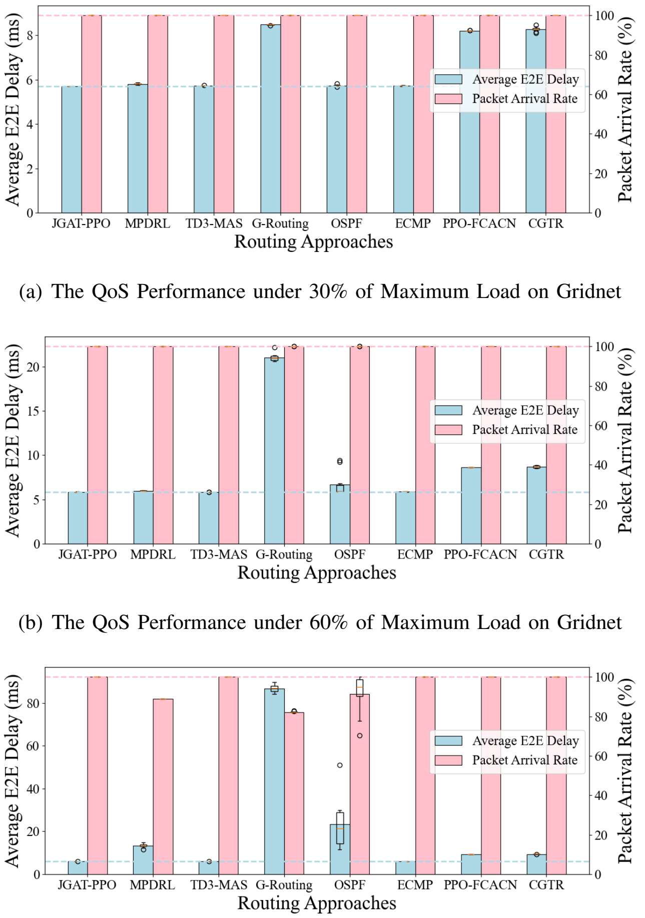
1) Convergence Analysis: Fig. 4 illustrates the agent-wise average rewards obtained by JGAT-PPO under 30%, 60%, and 90% of L_{max} on Gridnet, BtNorthAmerica, and Dfn with 10 independent runs. To capture changes under different loads accurately, the values of average rewards in Fig. 4 are normalized, and bias is introduced. Initially, JGAT-PPO employs a random policy, leading to low reward values at the onset of training. Subsequently, JGAT-PPO iteratively refines the policy by continuously interacting with the environment, which is reflected in the fact that the average rewards in Fig. 4 progressively increase and eventually converge. In addition, the standard deviation of average rewards exhibits no significant fluctuations during all experiments, demonstrating remarkable adaptability and stability of JGAT-PPO throughout the training process.

The results in Fig. 4 further reveal some interesting effects of different network conditions on the convergence performance of the proposed approach. From the perspective of reward value jitter, the average reward jitter on Gridnet is higher at 30% and 60% of $L_{\max}$ compared to 90% of $L_{\max}$, as shown in Fig. 4(a). In contrast, as shown in Fig. 4(b)-4(c), the average reward jitter is higher at 90% of $L_{\max}$ compared to 30% and 60% of $L_{\max}$ on BtNorthAmerica and Dfn. This may be due to the low loads on small topologies that make the influence of the random seed on the policy optimization process more significant. Conversely, high loads on large-scale topologies tend to cause more pronounced effects from the random seed. In this case, the average reward of JGAT-PPO gradually increases and eventually converges across all network conditions, demonstrating the ability of JGAT-PPO to learn and make decisions effectively under varying loads and topologies.

2) Comparison With Benchmark Approaches: Fig. 5–7 presents comparisons of the proposed routing approach with other benchmark approaches mentioned in Section IV-D in terms of QoS performance. The comparison includes two state-of-the-art DRL-based probabilistic packet forwarding approaches (i.e., PPO-FCACN and CGTR), three state-of-the-art DRL-based routing approaches (i.e., MPDRL, G-Routing, and TD3-MAS), as well as two conventional routing approaches (i.e., OSPF and ECMP). All routing approaches are evaluated on Gridnet, BtNorthAmerica, and Dfn at traffic intensities corresponding to 30%, 60%, and 90% of $L_{\max}$ to observe their performance under different loads across various topology scales. Without loss of generality, each approach independently runs 10 times under different random seeds for each topology and load.

For better demonstration, the packet arrival rate $ar(t) = 1 - lr(t)$ at time step t is used instead of the packet loss rate $lr(t)$ in Fig. 5–7. Namely, a higher packet arrival rate means a lower packet loss rate and a better QoS performance. For facilitating comparison with the QoS performance between JGAT-PPO and other routing approaches, the average E2E delay of JGAT-PPO is stressed by the blue dashed line, and the packet arrival rate of JGAT-PPO is emphasized by the red dashed line in Fig. 5–7.

On the small-scale Gridnet topology, the proposed JGAT-PPO achieves the best average E2E delay and packet arrival rate under all load conditions compared to all benchmark approaches, as shown in Fig. 5. At 30% and 60% of $L_{\max}$ on Gridnet, OSPF, and ECMP can be considered to approximate optimal routing performance as there is low traffic intensity and sufficient link resources. At this point, unconstrained probabilistic packet forwarding approaches (i.e., PPO-FCACN and CGTR) significantly lag in QoS performance at 30% and 60% of $L_{\max}$, although they achieve excellent QoS performance at 90% of $L_{\max}$. In contrast, the proposed JGAT-PPO achieves the best QoS performance under different experimental loads. This demonstrates that the constraint-aware probabilistic packet forwarding approach dramatically enhances the capability of probabilistic packet forwarding for routing optimization, especially under low load conditions on small-scale topologies.



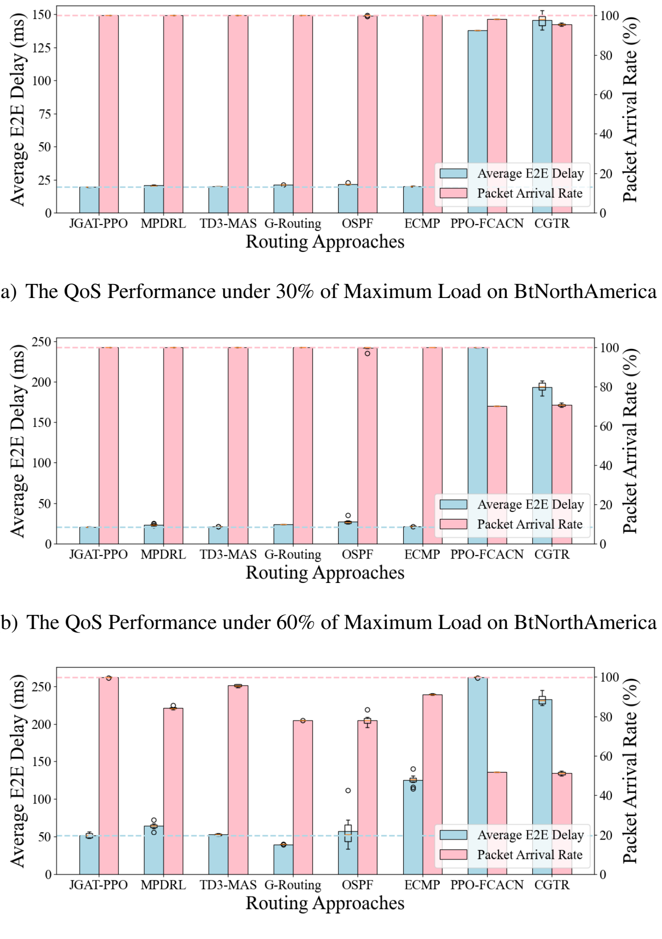
(a) The QoS Performance under 30% of Maximum Load on Gridnet

(b) The QoS Performance under 60% of Maximum Load on Gridnet

(c) The QoS Performance under 90% of Maximum Load on Gridnet

Fig. 5. The performance comparison of JGAT-PPO and other benchmark approaches under 30%, 60%, and 90% of maximum load on Gridnet.

On the medium-scale BtNorthAmerica topology, JGAT-PPO still performs the best QoS compared to other benchmark approaches, as shown in Fig. 6. However, unconstrained probabilistic packet forwarding approaches struggle to route effectively as the network scale increases, achieving the lowest packet arrival rate and the highest average E2E delay under all experimental loads. In contrast, JGAT-PPO continues to effectively optimize routing even with the increased network scale, exhibiting the highest packet arrival rate and competitive average E2E delay compared to other benchmark approaches across all loads. Additionally, JGAT-PPO shows significantly better QoS performance than other benchmark approaches at 90% of $L_{\max}$, demonstrating the effectiveness of the proposed approach in routing optimization, particularly under overload

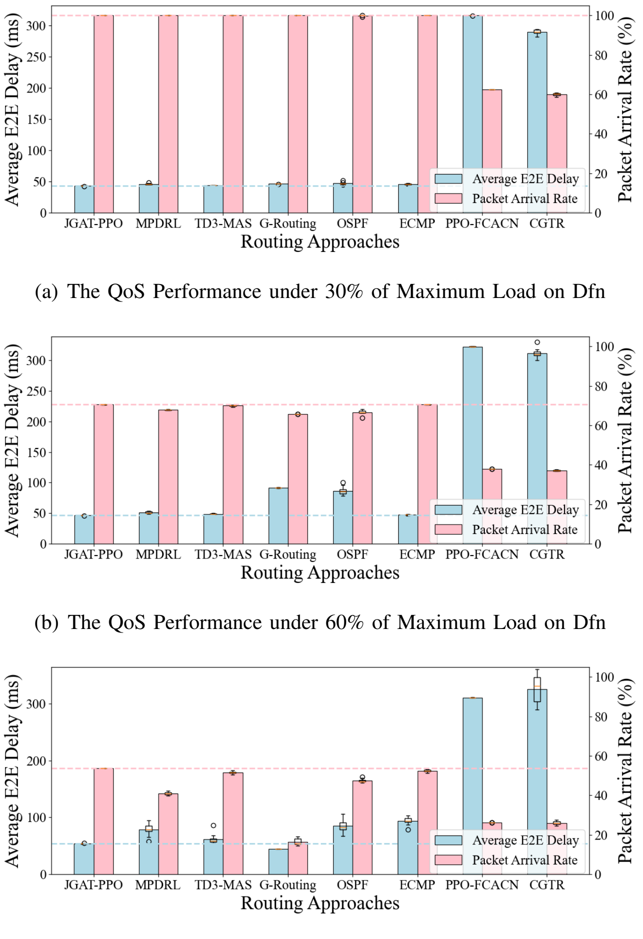


(a) The QoS Performance under 30% of Maximum Load on BtNorthAmerica

(b) The QoS Performance under 60% of Maximum Load on BtNorthAmerica

(c) The QoS Performance under 90% of Maximum Load on BtNorthAmerica

Fig. 6. The performance comparison of JGAT-PPO and other benchmark approaches under 30%, 60%, and 90% of maximum load on BtNorthAmerica.



(a) The QoS Performance under 30% of Maximum Load on Dfn

(b) The QoS Performance under 60% of Maximum Load on Dfn

(c) The QoS Performance under 90% of Maximum Load on Dfn

Fig. 7. The performance comparison of JGAT-PPO and other benchmark approaches under 30%, 60%, and 90% of maximum load on Dfn.

conditions.

On the large-scale Dfn topology, the two unconstrained probabilistic packet forwarding approaches still lag significantly behind as the network scale increases, as shown in Fig. 7. In contrast, the proposed constraint-aware packet forwarding approach continues to exhibit the best QoS performance among the benchmark approaches. Unlike previous cases, the further increase in topology scale causes JGAT-PPO to experience packet loss as the load reaches 60% of $L_{\max}$. However, JGAT-PPO still maintains the highest packet arrival rate and the lowest average E2E delay compared to other benchmark approaches. With the load on Dfn increase to 90% of $L_{\max}$, only TD3-MAS can maintain a packet arrival rate similar to JGAT-PPO but with higher average E2E delay, while all other benchmark approaches fall far behind. The experimental results shown in Fig. 7 further demonstrate that JGAT-PPO still possesses excellent capabilities for routing optimization when facing large-scale topology.

It is worth noting that G-Routing exhibits a lower average E2E delay than JGAT-PPO but a significantly lower packet arrival rate under 90% of $L_{\max}$ on BtNorthAmerica and Dfn, as shown in Fig. 6(c) and Fig. 7(c). It can be explained that a few packets can reach the destination node through idle links with lower E2E delay when network links are not fully congested. Network traffic gradually saturates as the number of packets increases, leading to packet loss. The network simulation framework in the experiments only calculates the average E2E delay for packets that successfully reach their destination node, as mentioned in Section II-B. Therefore, the successfully delivered packets still maintain a relatively low average E2E delay even though the packet arrival rate decreases, which results in a lower average E2E delay calculation. Most importantly, the optimization goal in the experiments is to maintain a low average E2E delay while ensuring a high packet arrival rate. Thus, despite G-Routing showing a lower average E2E delay than JGAT-PPO, the significantly lower packet arrival rate of G-Routing compared to JGAT-PPO is sufficient to indicate that G-Routing performs worse in routing optimization than JGAT-PPO.

In summary, the experimental results demonstrate the effectiveness of routing optimization and robustness when facing topologies and loads of varying scales of the proposed constraint-aware probabilistic packet forwarding approach. The proposed JGAT-PPO outperforms all benchmark approaches in terms of better network QoS performance and the most competitive capability for routing optimization. Compared to two state-of-the-art but unconstrained probabilistic packet forwarding approaches, JGAT-PPO shows significant improvement in routing optimization under low and medium load conditions on the small-scale topology and maintains the best network QoS performance when facing increased network scale. In contrast, unconstrained probabilistic packet forwarding approaches fall far behind. The experimental results fully illustrate the effectiveness and necessity of the constraint-aware probabilistic packet forwarding approach, showcasing more substantial routing optimization capabilities and robustness when facing changing topologies and loads compared to all benchmark approaches.

V. Conclusion

This paper has proposed a novel constraint-aware probabilistic packet forwarding approach called JGAT-PPO based on deep reinforcement learning. By leveraging an innovative joint graph attention network structure, JGAT-PPO effectively utilizes information of topologies for routing optimization and minimizes routing loops through the proposed probabilistic packet forwarding protocol to ensure routing safety. Additionally, this paper has designed and implemented a packet-level network simulation framework based on knowledge-defined networking, enabling the comparison of probabilistic packet forwarding approaches with other DRL-based and conventional routing approaches. Comprehensive experiments conducted within the implemented network simulation framework demonstrate that JGAT-PPO exhibits the most competitive quality-of-service performance compared to two state-of-the-art probabilistic packet forwarding approaches, three state-of-the-art routing approaches, and two conventional routing approaches across topologies and loads of varying scales. In particular, the constraints of JGAT-PPO on probabilistic packet forwarding show significant improvements in network quality-of-service performance and robustness compared to two unconstrained state-of-the-art probabilistic packet forwarding approaches. Additionally, we have noted cognitive routing and look forward to exploring the integration of probabilistic forwarding and cognitive routing in future works [28].

References

[1] J. Moy, "OSPF version 2," *IETF, RFC 2328*, 1998.

[2] C. Hopps, "Analysis of an equal-cost multi-path algorithm," *IETF, RFC 2992*, 2000.

[3] Y. Xiao, J. Liu, J. Wu, and N. Ansari, "Leveraging deep reinforcement learning for traffic engineering: A survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2064–2097, 2021.

[4] V. Tumas, S. Rivera, D. Magoni, and R. State, "Probabilistic edge multicast routing for the XRP network," in *2022 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2022, pp. 5129–5134.

[5] F. Wang, F. Yan, X. Xue, B. Liu, L. Zhang, Q. Zhang, X. Xin, and N. Calabretta, "Traffic load balancing based on probabilistic routing in data center networks," in *2020 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2020, pp. 1–3.

[6] T. Mori, K. Hirata, and M. Yamamoto, "Content-oriented probabilistic routing with measured RTT," in *2016 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2016)*. IEEE, 2016, pp. 1–6.

[7] Y. Xiao, H. Yu, Y. Yang, Y. Wang, J. Liu, and N. Ansari, "Adaptive joint routing and caching in knowledge-defined networking: An actor-critic deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2024.

[8] X. Li, Y. Xiao, S. Liu, X. Lu, F. Liu, W. Zhou, and J. Liu, "GAPPO-A graph attention reinforcement learning based robust routing algorithm," in *2023 IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2023, pp. 1–7.

[9] J. Chen, Y. Xiao, G. Lin, G. He, F. Liu, W. Zhou, and J. Liu, "Deep reinforcement learning based dynamic routing optimization for delay-sensitive applications," in *2023 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2023, pp. 5208–5213.

[10] Y. Wang, Y. Xiao, Y. Song, J. Zhou, and J. Liu, "Deep reinforcement learning based probabilistic cognitive routing: An empirical study with OMNeT++ and P4," in *2023 International Conference on Network and Service Management (CNSM)*. IEEE, 2023, pp. 1–7.

[11] J. Zhou, Y. Song, X. Li, W. Zhou, and J. Liu, "Link2Link: A robust probabilistic routing algorithm via edge-centric graph reinforcement learning," in *2024 International Conference on Network and Service Management (CNSM)*. IEEE, 2024, pp. 1–7.

[12] X. Li, J. Li, Y. Xiao, S. Liu, and J. Liu, "CGTR: Leveraging contrastive learning and graph transformer for deep reinforcement learning based robust routing," in *2024 IEEE International Conference on Communications (ICC)*. IEEE, 2024, pp. 472–478.

[13] K. Lei, J. Yuan, and J. Wang, "Mdpf: An ndn probabilistic forwarding strategy based on maximizing deviation method," in *2015 IEEE global communications conference (GLOBECOM)*. IEEE, 2015, pp. 1–7.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[16] Y. Xiao, Y. Yang, H. Yu, and J. Liu, "Scalable qos-aware multipath routing in hybrid knowledge-defined networking with multiagent deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 11, pp. 10 628–10 646, 2024.

[17] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[18] J. Chen, Y. Xiao, and G. Lin, "RL4NET++: A packet-level network simulation framework for DRL-based routing algorithms," in *2023 IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*. IEEE, 2023, pp. 248–253.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[20] L. Mészáros, A. Varga, and M. Kirsche, "Inet framework," *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, pp. 55–106, 2019.

[21] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, 2010.

[22] P. Hintjens, *ZeroMQ: Messaging for Many Applications*. Newton, MA, USA: O'Reilly Media, Inc., 2013.

[23] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[25] Q. He, Y. Wang, X. Wang, W. Xu, F. Li, K. Yang, and L. Ma, "Routing optimization with deep reinforcement learning in knowledge defined networking," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1444–1455, 2024.

[26] H. Wei, Y. Zhao, and K. Xu, "G-routing: Graph neural networks-based flexible online routing," *IEEE Network*, vol. 37, no. 4, pp. 90–96, 2023.

[27] S. Barzegar, M. Ruiz, and L. Velasco, "Autonomous flow routing for near real-time quality of service assurance," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2504–2514, 2024.

[28] Y. Xiao, J. Li, J. Wu, and J. Liu, "On design and implementation of reinforcement learning based cognitive routing for autonomous networks," *IEEE Communications Letters*, vol. 27, no. 1, pp. 205–209, 2022.