GNN-Based Hierarchical Deep Reinforcement Learning for NFV-Oriented Online Resource Orchestration in Elastic Optical DCIs

Baojia Li and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—Network function virtualization (NFV) in elastic optical datacenter interconnections (EO-DCIs) enables flexible and timely deployment of network services. However, as the service provisioning of virtual network function service chains (vNF-SCs) in an EO-DCI needs to orchestrate the allocations of IT resources in datacenters (DCs) and spectrum resources on fiber links dynamically, it is a complex and challenging problem. In this work, we model the problem as a Markov decision process (MDP), and propose a hierarchical deep reinforcement learning (DRL) model based on graph neural network (GNN), namely, HRLOrch, to tackle it. To ensure its universality and scalability, we design the policy neural network (NN) in HRLOrch based on a GNN. As the GNN-based policy NN can operate on the graph-structured network state of an EO-DCI directly, it can adapt to an arbitrary EO-DCI topology without any structural changes. Then, through analysis, we find that the EO-DCI is a sparse reward environment if we want to train a DRL model to minimize the blocking probability of vNF-SCs in it directly. To address this issue, we design a hierarchical DRL with lower-level and upper-level models to improve the convergence performance of training. Specifically, we make the lower-level DRL optimize the provisioning scheme of each vNF-SC to minimize its resource usage, while the upper-level one coordinates the provisioning of all the active vNF-SCs to minimize the overall blocking probability. Hence, the lower-level and upper-level DRL models operate cooperatively in the training to optimize the dynamic provisioning of vNF-SCs. Our simulations demonstrate the universality and scalability of HRLOrch, and confirm that it can outperform the existing algorithms for vNF-SC provisioning in an EO-DCI.**

***Index Terms*—Network function virtualization (NFV), service function chain, datacenter interconnection (DCI), elastic optical network (EON), graph neural network (GNN), deep reinforcement learning (DRL), network automation.**

I. Introduction

NOWADAYS, the growth of 5G, high-definition and immersive video applications, and Big Data analytics has generated the demands for flexible, timely and cost-effective network service deployment [1], [2]. Hence, service providers (SPs) have to upgrade their service provisioning strategy from relying on dedicated middleboxes to counting on network function virtualization (NFV) [3], [4]. Specifically, NFV deploys a network service by instantiating virtual network functions (vNFs) on general-purpose servers and storages, and leveraging commodity switches to interconnect the vNFs [5]. In other words, with NFV, network services can be realized timely, dynamically and flexibly by forming vNF service chains (vNF-SCs) [6], vNF trees [7], and generic vNF graphs [8], such that the tradeoff between the quality-of-service (QoS) and the cost and complexity of service deployment can be better optimized.

Due to the abundance of commodity servers, storage and switches in datacenters (DCs), NFV-based service deployment is usually considered in the intra-DC and DC interconnection (DCI) networks [9], [10]. The rationale of considering DCIs is to leverage geographically-distributed DCs for improving SPs' performance on service coverage, latency, and availability [11], [12]. Meanwhile, as the NFV-oriented resource orchestration in a DCI needs to further tackle how to manage the spectrum resources in an optical network [13], [14], it is intrinsically more complex than its counterpart in an intra-DC network. Despite the huge bandwidth capacity of fiber links, optical networks now can achieve fine-grained, adaptive and application-aware spectrum allocation in the physical layer, with the momentum gained from flexible-grid elastic optical networking (EON) [15]–[18]. Therefore, in an elastic optical DCI (EO-DCI), lightpaths can be established with a spectrum allocation granularity of 12.5 GHz or even narrower, to adapt to the bandwidth requirements of NFV-oriented resource orchestration seamlessly.

Note that, NFV-oriented resource orchestration in EO-DCIs is a complex and challenging problem even for the simplest network service deployment (*i.e.*, forming vNF-SCs). This is because, to assemble one vNF-SC in an EO-DCI, the SP needs to deploy new or reuse existing vNFs in suitable DCs (*i.e.*, the vNF deployment problem) and set up lightpaths in the DCI with routing and spectrum assignment (RSA) to connect the vNFs in sequence (*i.e.*, the RSA problem) [19]. Hence, even though previous studies have formulated integer linear programming (ILP) models and designed time-efficient heuristics to address it [19]–[22], the resource orchestration to provision vNF-SCs in an EO-DCI still deserves to be revisited, especially for dynamic and large-scale network environments.

Manuscript received June 20, 2021; revised October 24, 2021; accepted November 3, 2021. Date of publication November 9, 2021; date of current version February 16, 2022. This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1806400, in part by NSFC Project under Grant 61871357, in part by the SPR Program of CAS under Grant XDC02070300, and in part by the Fundamental Funds for Central Universities under Grant WK3500000006. *(Corresponding author: Zuqing Zhu.)*

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: libaojia@mail.ustc.edu.cn; zqzhu@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2021.3125974.

Digital Object Identifier 10.1109/JLT.2021.3125974

The online resource orchestration to provision vNF-SCs in an EO-DCI can be modeled as a Markov decision process (MDP). Specifically, the MDP model considers resource utilization in the EO-DCI (*i.e.*, the spectrum utilizations on fiber links and IT resource usages on DCs) as the network state, and defines the provisioning/removing of a vNF-SC as an action. Therefore, dynamic transactions among the network states can be captured accurately. Recently, deep reinforcement learning (DRL) has been regarded as a promising technique to tackle the complex and dynamic optimizations that can be modeled as MDPs [23], for realizing timely and intelligent decision making [24]. Specifically, a DRL model leverages one or more agents, each of which contains a deep neural network (DNN), to interact with time-variant environment and learn the strategy to address each environment state with the best decision [23].

Motivated by the aforementioned advantages, latest studies have designed DRL models to solve the subproblems related to provisioning vNF-SCs in an EO-DCI, *e.g.*, vNF deployment [25] and RSA [26]. Promisingly, the DRL models achieved better performance than existing heuristics, with comparable time-efficiency. However, these models were not crafted for the whole problem of provisioning vNF-SCs in an EO-DCI.

Moreover, the existing DRL models for provisioning vNF-SCs still have a few drawbacks. First of all, the DNN in a DRL agent operates on Euclidean space, while the network state of an EO-DCI is in graph structure. Hence, the DNN cannot process the graph-structured data of the network state effectively, and certain important features cannot be extracted. Secondly, as the provisioning scheme of a vNF-SC involves the deployment of required vNFs and the RSA of related lightpaths, it can only be modeled with many decision variables. If we directly design the action space of a DRL agent based on the decision variables (as the studies in [25], [26] did), the action space will be extremely large, which can make the DRL training difficult to converge. Finally, because the network state of an EO-DCI is relatively complicated, a DRL agent can face the challenge of sparse extrinsic reward. Nevertheless, the existing DRL models only followed the generic principle of DRL, but did not incorporate specific designs to deal with the sparse reward environment of an EO-DCI.

In this work, we propose a hierarchical DRL model based on graph neural network (GNN), namely, HRLOrch, to tackle the online resource orchestration for provisioning vNF-SCs in an EO-DCI. We design the policy neural network (NN) of HRLOrch based on GNN, and thus it can operate directly on the graph-structured data about the EO-DCI to understand and extract the complex features buried in it effectively [27]. Then, to address the sparse reward environment of the EO-DCI, we introduce a hierarchical model for DRL. This model optimizes the provisioning of vNF-SCs from both the micro and marco perspectives, by including the lower-level and upper-level DRL models. The lower-level DRL model obtains the provisioning scheme of each vNF-SC request to minimize its resource usage, while the upper-level one coordinates the provisioning schemes of all the active vNF-SCs such that the overall blocking probability can be minimized. Meanwhile, in order to improve training efficiency, we design a hierarchical training scheme to collaborate the training processes of the lower- and upper-level DRL models. Finally, we carefully design the action spaces of the upper-/lower-level DRL models with reduced sizes, and architect a GNN-based policy NN to realize the mapping from network state to action directly. Extensive simulations are performed to evaluate HRLOrch in various network scenarios, and the results confirm that it can outperform the existing algorithms.

The rest of the paper is organized as follows. Section II surveys the related work. We describe the problem of online resource orchestration for provisioning vNF-SCs in an EO-DCI in Section III. The architecture and operation principle of HRLOrch are presented in Section IV, while the design of the GNN-based policy NN is elaborated in Section V. We evaluate the performance of HRLOrch with numerical simulations in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

Both NFV [3], [4] and network virtualization [28] are attractive virtualization technologies that were proposed to address the ossification of current Internet infrastructure. Previously, using network virtualization as the background, researchers have studied the virtual network embedding (VNE) problem intensively, in different types of networks and with various optimization objectives [29]–[32]. Specifically, VNE considers how to build multiple virtual networks (VNTs) over a shared substrate network (SNT), which, however, is fundamentally different from NFV-oriented resource orchestration [7]. This is due to the fact that VNE finalizes the topologies of VNTs before embedding them in the SNT, but how to route traffic in the VNTs is out of its scope since the traffic will be generated afterwards. On the other hand, NFV-oriented resource orchestration can only obtain the actual topology of each vNF-based network service after the embedding because multiple vNFs might share a same substrate node, while the traffic routing of the network service is predetermined.

Previously, the service provisioning of vNF-SCs has been considered for packet networks in numerous studies, *e.g.*, in [33]–[37]. Several ILP or mixed ILP (MILP) models have been formulated in [33]–[35] to get optimal solutions for small-scale problems, and approximation algorithms were also proposed to better balance the tradeoff between time-efficiency and solution-optimality. A few heuristics were designed and compared in [36], [37] to solve vNF-SC provisioning in large-scale and dynamic networks. Nevertheless, as these studies did not address the problem of RSA in optical DCIs, their proposals cannot be leveraged for vNF-SCs provisioning in EO-DCIs.

Considering an EO-DCI as the SNT, NFV-oriented resource orchestration has been investigated for provisioning vNF-SCs [19]–[22], vNF trees [7], and generic vNF graphs [8]. However, these studies generally followed the idea of first formulating ILP/MILP models to obtain exact solutions and then designing time-efficient heuristics to address large-scale problems, but they did not analyze the state transition of the MDP for provisioning vNF-SCs in an EO-DCI to optimize the provisioning schemes. This leaves us certain margin to further optimize the tradeoff between time-efficiency and solution-optimality with DRL. Note that, the two major subproblems of provisioning vNF-SCs in an EO-DCI, *i.e.*, the vNF deployment and RSA problems, have been addressed in [25] and [26], respectively, with DRL models. Nevertheless, the DRL models cannot work jointly to solve the two subproblems, and they bear scalability and universality issues, as explained in the previous section. In [38], [39], we leveraged DRL to predict future vNF-SC requests and adjust the duration of service cycles adaptively in an EO-DCI, but the developed DRL models did not address the resource orchestration for assembling vNF-SCs.

In this work, we propose a hierarchical DRL model to solve the provision of vNF-SCs in an EO-DCI, which leverages a well-crafted policy NN based on GNN [40]. GNN can directly operate on graph-structured data and has demonstrated good performance in dynamic optimizations related to networks. For instance, the authors of [40] leveraged GNN to achieve accurate prediction of delay and jitter in packet networks, and they proved that the advantages were achieved because of GNN modeling the relation among nodes and links better and having a comprehensive understanding of path information.

III. Problem Formulation and Modeling

In this section, we first describe the network model of provisioning vNF-SCs in an EO-DCI and define the problem of the resource orchestration for it, and then explain why the problem can be modeled as an MDP.

A. Network Model

We model the topology of an EO-DCI as a graph $G(V, E)$, where V represents the set of DC nodes and E denotes the set of fiber links to interconnect the DCs. Each DC node contains a DC and a bandwidth-variable optical cross-connect (BV-OXC), which are responsible for instantiating vNFs and establishing inter-DC lightpaths, respectively. For the DC on node $v \in V$, its IT resource capacity is C_v units, which can be used to deploy vNFs. The spectra on each fiber link $e \in E$ can be allocated according to the flexible-grid scenario [41], *i.e.*, the fiber link accommodates F 12.5-GHz frequent slots (FS').[1] To bridge the communication between two adjacent vNFs in a vNF-SC, we need to set up a lightpath if the vNFs are deployed on different DCs. The RSA scheme of the lightpath should comply with the spectrum contiguous, non-overlapping and continuous constraints [15], [42].

The DCs in the EO-DCI are assumed to support M types of vNFs, and an m-th type vNF ($m \in [1, M]$) consumes c_m units of IT resources. Note that, a vNF might be shared by multiple vNF-SCs if they all require the same type of vNF, and one vNF of type-m can only process the traffic of η_m vNF-SCs at most. Each vNF-SC consists of a series of vNFs and can be modeled as $SC = \{\delta_1, \delta_2, \ldots, \delta_K\}$, where δ_k is k-th vNF in it. Then, a vNF-SC request arrived at time t is $\mathcal{R}^t(o^t, d^t, b^t, SC^t, \tau_a^t, \tau_h^t)$, where o^t and d^t are the source and destination DC nodes, respectively, b^t is its bandwidth demand in Gb/s, SC^t is the required vNF-SC, and τ_a^t and τ_h^t are the arrival and holding time, respectively. To serve a vNF-SC request $\mathcal{R}^t$, we need to solve the deployment of required vNFs and the RSA of related lightpaths. For the vNF deployment, we find suitable DCs to instantiate new vNFs or reuse the existing ones on them, such that all the vNFs in SC^t are covered. Then, we calculate the RSA schemes of the lightpaths to connect the vNFs in sequence, to satisfy the bandwidth demand b^t.

[1]For simplicity, we assume that each FS provides a capacity of 12.5 Gb/s.

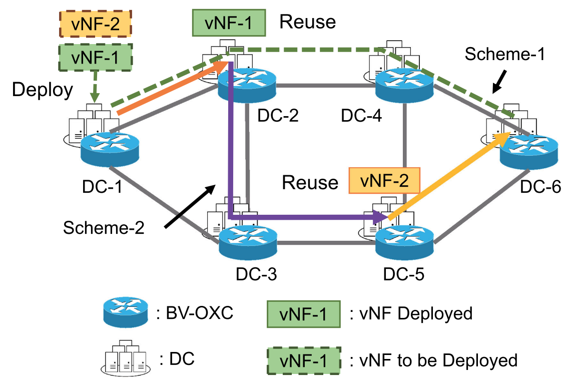


Fig. 1. Provisioning a vNF-SC request in one EO-DCI with heuristics.

B. Assembling vNF-SCs in an EO-DCI

As solving ILP/MILP models is usually time-consuming and thus cannot adapt to the timing requirement of dynamic service provisioning, we normally can only count on time-efficient heuristics to provision vNF-SC requests in a large-scale EO-DCI. However, the dilemma is that a heuristic usually cannot guarantee the performance gap to the optimal solution. In other words, a heuristic might operate well in certain situations but would have difficulties to address the others.

Fig. 1 gives an example on provisioning a vNF-SC request in an EO-DCI with different heuristics. The request has its source and destination DC nodes as *DC*-1 and *DC*-6, respectively, the required vNF-SC is *DC*-1→*vNF*-1→*vNF*-2→*DC*-6, and its bandwidth demand is 20 Gb/s, which can be accommodated with 3 FS' (including one guard-band FS). Here, *vNF*-1 means an instance of the first type vNF, and so on. We leverage two well-known heuristics to serve the vNF-SC request. The first one tries to minimize spectrum usage, and thus it sets up a lightpath from *DC*-1 to *DC*-6 with the shortest path routing (*i.e.*, the green dash line in Fig. 1) and deploys new *vNF*-1 and *vNF*-2 in *DC*-1. The second one tries to minimize IT resource usage by reusing the existing vNFs as many as possible. Hence, it reuses the existing *vNF*-1 and *vNF*-2 in *DC*-2 and *DC*-5, respectively, and establishes three lightpaths (*i.e.*, the solid lines in Fig. 1) to form the vNF-SC.

By comparing the service provisioning schemes from the two heuristics, we can see that the former uses fewer spectra and bandwidth-variable transponders (BV-Ts) but more IT resources than the latter, and *vice versa*. Therefore, which of the heuristics will provide smaller blocking probability really depends on which type of resources (*i.e.*, spectrum or IT resources) is scarce in the concerned EO-DCI. Nevertheless, in a dynamic network environment, both the resource usages and the resource demands of vNF-SC requests change over time, which suggests that there might not be a constant winner. In other words, a deterministic heuristic could not provision vNF-SC requests adaptively according to the actual network state of the EO-DCI. This motivates us to propose HRLOrch, which leverages a GNN-based hierarchical DRL model to solve the resource orchestration for assembling vNF-SCs in an EO-DCI.

C. *Modeling Dynamic vNF-SC Provisioning as an MDP*

In this work, we consider the dynamic scenario of vNF-SC provisioning, where the requests arrive and expire on-the-fly and are served one by one in the order of their arrival time. The optimization objective is to minimize the overall blocking probability. To capture the dynamic transitions among network states during the service provisioning, we leverage an MDP. Specifically, we describe the MDP with a tuple $<S, A, R, P>$, where S and A are the state and action spaces, respectively, R is the immediate reward function, and P is the distribution of state transition probabilities.

The state at time t (*i.e.*, $s_t \in S$) consists of two elements, which are a newly-arrived vNF-SC request $\mathcal{R}^t$ and the current network state G_t. The information of request $\mathcal{R}^t$ is represented by a feature vector, and we model the network state as graph-structured data $G_t(V, E)$ based on the topology of the EO-DCI $G(V, E)$. In $G_t(V, E)$, each node $v \in V$ corresponds to a DC node v in the EO-DCI. We record the features of node v in a vector x^v, which includes $M + 1$ elements to denote the available IT resources and remaining traffic processing capacities of the existing vNFs on node v, respectively. Each link $e \in E$ in $G_t(V, E)$ denotes a fiber link in the EO-DCI, and its feature vector x^e contains F elements, each of which tells whether the corresponding FS on link e is used or not.

The action at time t (*i.e.*, $a_t \in A$) is defined as a feasible provisioning scheme of the request $\mathcal{R}^t$. As the objective of vNF-SC provisioning is to minimize the blocking probability, we define the reward as $r_t = 1$ if $\mathcal{R}^t$ gets served successfully, and $r_t = -1$, otherwise. A state transition of the MDP can be denoted as (s_t, a_t, r_t, s_{t+1}). This means that $\mathcal{R}^t$ is served with the provisioning scheme in a_t at state s_t, which makes the network state change to s_{t+1} and obtains a reward r_t.

Although the MDP above can precisely model the dynamic service provisioning of vNF-SCs in an EO-DCI, architecting a DRL model directly based on it will face the following challenges. Note that, most of practical optical networks operate with the blocking probability less than 10% even for lightpath setup [43], and emerging network paradigms [44] can have even more stringent requirements on blocking probability. Hence, it will be reasonable to assume that most of the vNF-SC requests (*i.e.*, more than 90% of them) can be served successfully. This means that in different states of the EO-DCI, the rewards obtained by applying various actions are the same. Hence, the EO-DCI is a sparse reward environment, which makes it difficult to optimize the policy of action selection in online training. Secondly, in a large-scale EO-DCI, we need to use many decision variables to describe the provisioning scheme of a vNF-SC, and thus the action space will be extremely large and not universal,

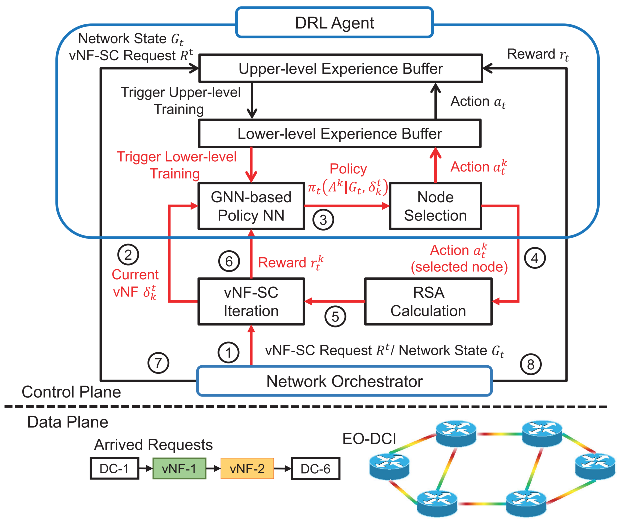


Fig. 2. Overall architecture and operation principle of HRLOrch.

if we design the action space directly based on the decision variables. Finally, as both the state and action of the MDP are relatively complicated, it will be hard to obtain the mapping from the state space to the action space, *i.e.*, the policy of action selection.

In the next section, we will propose a GNN-based hierarchical DRL model to address these challenges. Specifically, in the hierarchical DRL model, the MDP mentioned above becomes the upper-level MDP, and a lower-level MDP is introduced to assist it. The lower-level MDP optimizes the provisioning of each vNF-SC to minimize its blocking probability, and provides a fine-tuned policy to the upper-level MDP after each vNF-SC provisioning, to help minimizing the overall blocking probability. Hence, the hierarchical DRL model can effectively address the issues caused by the sparse reward environment.

IV. HRLORCH: HIERARCHICAL DRL MODEL FOR VNF-SC PROVISIONING

This section presents the overall architecture of our proposed HRLOrch, and explains its operation principle.

A. *System Architecture and Operation Principle*

Our proposed GNN-based hierarchical DRL model consists of an upper-level MDP and a lower-level MDP. The upper-level MDP is used to precisely model the dynamic provisioning of vNF-SCs and to minimize the overall blocking probability from the macro perspective, while the lower-level MDP is designed for tackling the provisioning of each vNF-SC (*i.e.*, optimizing vNF-SC provisioning from the micro perspective). Meanwhile, we build a GNN-based policy NN and design a hierarchical training scheme for our DRL model to improve the convergence and performance of its training.

Fig. 2 shows the overall architecture of HRLOrch and explains its operation principle. Specifically, working as an intelligent assistant to the network orchestrator (NOrch), HRLOrch resides in the control plane, and calculates vNF-SC provisioning schemes based on the information that the NOrch collects from the EO-DCI. Specifically, as illustrated in Fig. 2, the operation of HRLOrch includes 8 steps.

When a new vNF-SC request R^t arrives, the NOrch records the current state of the EO-DCI as graph-structured data $G_t(V, E)$, denotes the information about R^t as a feature vector, and sends them to the vNF-SC iteration module (**Step 1**). Here, $G_t(V, E)$ is graph-structured data, where the features of each DC node $v \in V$ and each fiber link $e \in E$ are represented as x^v and x^e, respectively. To make the action space more universal and reduce its size, we redefine each action as to determine the placement of a vNF in the required vNF-SC SC^t of $\mathcal{R}^t$. Therefore, we introduce the vNF-SC iteration module to check each vNF in SC^t in sequence and send it to the GNN-based policy NN (**Step 2**).

Then, the GNN-based policy NN takes the network state (G_t) and the vNF to be deployed (δ_k^t) as the input, and generates a probability distribution over all the DC nodes, *i.e.*, the policy of action selection $\pi(V|G_t, \delta_k^t)$ (**Step 3**). Next, the hierarchical DRL agent leverages the node selection module to choose a DC node to be deploy vNF δ_k^t by sampling the policy $\pi(V|G_t, \delta_k^t)$ (**Step 4**). After that, the RSA of the lightpath (if there needs one) from vNF δ_{k-1}^t to vNF δ_k^t is obtained by the RSA calculation module[2] (**Step 5**). Here, the separate steps are just for the convenience of explaining the operation of HRLOrch clearly. We actually design HRLOrch to optimize vNF deployment and lightpath setup jointly, because its action considers their schemes together.

Based on the status of node selection and lightpath setup, a lower-level reward r_t^k can be calculated to indicate the performance of the current action a_t^k (**Step 6**). If both the node selection and lightpath setup are successful, the hierarchical DRL agent proceeds to the next vNF in SC^t, *i.e.*, repeating the procedure of **Steps 2–l6**. Otherwise, if the DRL agent cannot find a feasible scheme for either of them, the vNF provisioning would be considered as failed and the vNF-SC is marked as blocked. After all the vNFs in SC^t have been served or any of them cannot be provisioned due to insufficient resources in the EO-DCI, we record the whole process as training samples in the lower-level experience buffer. The training samples will then be used to train the GNN-based policy NN to minimize the resource usage of each vNF-SC request.

Meanwhile, the whole service provisioning process is regarded as a training sample in the upper-level experience buffer, which is denoted as $<(G_t, \mathcal{R}^t), a_t, r_t, (G_{t+1}, \mathcal{R}^{t+1})>$ (**Steps 7–8**). The objective of the upper-level training is to minimize the overall blocking probability. Here, the upper-level reward r_t indicates whether $\mathcal{R}^t$ gets provisioned successfully, the action $a_t = \{a_t^k, k \in [1, K^t]\}$ is a vector that contains the selected DC nodes for all the vNFs in SC^t, where K^t is the number of vNFs in SC^t. Note that, to ensure the stability of the GNN-based policy NN, we trigger the lower-level training first, while the upper-level training will not be started until the lower-level training has converged.

[2]Note that, if vNFs δ_{k-1}^t and δ_k^t are deployed on the same DC node, we do not need to set up a lightpath between them. Meanwhile, for a vNF-SC request $\mathcal{R}^t$ with $SC^t = \{\delta_1^t, \delta_2^t, \dots, \delta_{K^t}^t\}$, we can denote its source o^t and destination d^t as vNFs δ_0^t and $\delta_{K^t+1}^t$, respectively, for simplicity.

B. Hierarchical Training of DRL Agent

The objective of the hierarchical DRL agent is straightforward, *i.e.*, to minimize the overall blocking probability of vNF-SC requests. Hence, the training process of the DRL agent needs to achieve this goal by updating the parameters of the GNN-based policy NN. However, as the EO-DCI is a sparse reward environment for vNF-SC provisioning, it would be difficult to train the DRL agent directly. Therefore, as shown in Fig. 2, we design a hierarchical training scheme, which introduces two training processes, *i.e.*, the lower-level and upper-level ones, and makes them collaborate with each other. We define the objective of the lower-level training process as to minimize the resource usage of each vNF-SC request, while the objective of the upper-level one is still to minimize the overall blocking probability. The rationale behind this design is that reducing the resource usage of each vNF-SC request will generally be beneficial for reducing the blocking probability.

Specifically, the hierarchical training runs as follows. We first trigger the lower-level training to let the DRL agent learn how to provision each vNF-SC request with the smallest resource usage, by updating the parameters of the GNN-based policy NN. Then, we start the upper-level training to further optimize the parameters of the GNN-based policy NN, such that the overall blocking probability can be minimized adaptively in a dynamic network environment. The benefits of this design are two-fold: 1) because the lower-level training will not encounter the issue of sparse reward environment and have a relatively small action space, it converges fast, and 2) the lower-level training provides a pre-trained GNN-based policy NN to the upper-level one, and thus accelerates it effectively. The details of the training processes are:

1) Lower-Level Training: For each vNF-SC request $\mathcal{R}^t$, the DRL agent iteratively uses the GNN-based policy NN to generate node selection policy and determines the placement (a_t^k) of each vNF in the vNF-SC SC^t. After processing the vNF, we get a reward r_t^k, which indicates the resource usage due to placing the vNF according to a_t^k. Then, the node selection result is recorded as a training sample $<(G_t, \delta_k^t), a_t^k, r_t^k, (G_t, \delta_{k+1}^t)>$ in the lower-level experience buffer. When all the vNFs in SC^t have been processed, the total resource usage of the provisioning scheme for $\mathcal{R}^t$ is

$$J^{\text{lower}} = \sum_{k=1}^{K^t} r_t^k. \tag{1}$$

As the reward ($r_t^{K^t}$) of the last vNF in SC^t (*i.e.*, $\delta_{K^t}^t$) considers the spectrum usage of two lightpaths ($\delta_{K^t-1}^t \to \delta_{K^t}^t$ and $\delta_{K^t}^t \to d^t$), the lower-level training optimizes the GNN-based policy NN by calculating the gradient of J^{lower}.

2) Upper-Level Training: For vNF-SC request $\mathcal{R}^t$, the DRL agent also obtains the current network state G_t and uses the GNN-based policy NN to get a provisioning scheme. This is the action considered in the upper-level training ($a_t = \{a_t^k, k \in [1, K^t]\}$). Then, a reward r_t can be obtained based on whether

$\mathcal{R}^t$ can be served with the provisioning scheme. Next, the network state changes to G_{t+1}, and we will check the next vNF-SC request $\mathcal{R}^{t+1}$. The aforementioned operations generate a training sample $<(G_t, \mathcal{R}^t), a_t, r_t, (G_{t+1}, \mathcal{R}^{t+1})>$, for being stored in the upper-level experience buffer. When the DRL agent has collected enough number of upper-level training samples, which is pre-defined as M, the upper-level training is triggered to minimize the overall blocking probability, which can be equivalently quantified as

$$J^{\text{upper}} = \sum_{i=0}^{M-1} r_{t+i}. \tag{2}$$

Then, we can leverage the policy gradient of J^{upper} to further optimize the GNN-based policy NN in the upper-level training.

C. Hierarchical DRL Model

Finally, we elaborate on the design of the hierarchical DRL model. As the upper-level DRL tries to optimize the provisioning of vNF-SC requests in an EO-DCI from the macro perspective, it can leverage the MDP model in Section III-C, which defines its state, action and reward. On the other hand, the MDP of the lower-level DRL is defined as follows.

1) State: For the lower-level DRL, the state at time t ($s_t \in S$) contains three elements, which are the current network state $G_t(V, E)$, a vNF (δ_k^t) to be deployed for the vNF-SC request $\mathcal{R}^t$, and a DC node set $\tilde{V}_k^t$. Here, $G_t(V, E)$ is the graph-structure data about the concerned EO-DCI $G(V, E)$ at time t, and $\tilde{V}_k^t$ includes all the DC nodes selected for the vNFs before δ_k^t in vNF-SC SC^t and the source node of $\mathcal{R}^t$.

2) Action: As we have explained before, the task of the lower-level DRL is to select the DC node for a pending vNF. Hence, its action space is defined as all the nodes in $G(V, E)$, and an action is to select a node from V.

3) Reward: The objective of the lower-level DRL is to minimize the resource usage of each vNF-SC. Hence, we define the immediate reward of each action as

$$r_t^k = \begin{cases} -\alpha, & \text{vNF is blocked,} \\ -\beta \cdot \psi_k - \gamma \cdot \phi_k, & \text{vNF is provisioned,} \end{cases} \tag{3}$$

where ψ_k indicates the IT resource usage for provisioning vNF δ_k^t, ϕ_k denotes the hop-count of the lightpath(s) that are set up to serve the current vNF δ_k^t, and α, β, and γ are positive coefficients. As $-\alpha$ is the value of the instant reward for the case in which the vNF is blocked, we should make sure that it is properly set to avoid such a case in the future. Meanwhile, since the IT resource usage ψ_k and hop count of the lightpath(s) ϕ_k use significantly different value ranges, β and γ are used to normalize their values. Note that, if δ_k^t can be served by reusing an existing vNF, we have $\psi_k = 0$, and if vNFs δ_{k-1}^t and δ_k^t are deployed on a same node, we do not need to set up a lightpath to connect them.

V. GNN-Based Policy Neural Network

In this section, we discuss the design and optimization of the GNN-based policy NN in our hierarchical DRL model.

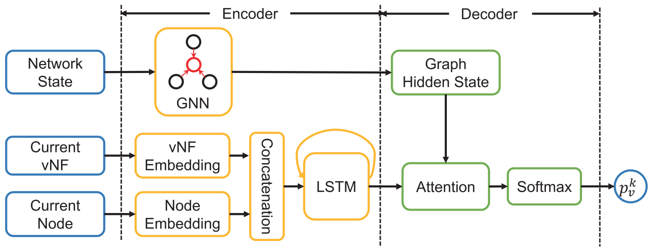


Fig. 3. Structure of our proposed GNN-based policy NN.

A. Structure of GNN-Based Policy Neural Network

In our hierarchical DRL model, the lower-level DRL needs to determine where to place each vNF in a pending vNF-SC in sequence, which can be considered as a series of sequential tasks. Meanwhile, we know that the encoder-decoder model [45] has relatively good performance on sequential tasks. Hence, we design the GNN-based policy NN, which is a key component in our hierarchical DRL model for provisioning vNF-SCs in an EO-DCI, based on the encoder-decoder model. Fig. 3 shows the structure of the GNN-based policy NN, which consists of the encoder and decoder sections. The encoder section abstracts the features of the current network state and the context of the pending vNF-SC, while the decoder section decides where to place the current vNF based on the output of the encoder. The two sections are explained as follows.

1) Encoder: It includes the encoders to extract the graph-based features of the network state G_t and the context of the provisioning scheme of the current request $\mathcal{R}^t$, respectively.

Note that, the traditional DNNs in linear or convolutional structures cannot operate on the graph-structured G_t directly, because it would be difficult for them to analyze the relations among the nodes and links in a graph with handcrafted feature engineering [27]. This is the reason why we design the encoder for analyzing G_t based on GNN. The GNN-based encoder leverages three phases to analyze G_t, *i.e.*, message passing, message aggregation and feature embedding. In the message passing phase, each node in $G_t(V, E)$ collects the information about its neighbor nodes. Then, the node aggregates all the received information to obtain an aggregated feature vector in the message aggregation phase. Finally, the feature embedding phase maps the feature vectors to a hidden state with an NN.

The aforementioned process can be represented as

$$H_t = f(X_V, \mathcal{A}_G) = \sigma(\mathcal{A}_G \cdot X_V \cdot W), \tag{4}$$

where X_V is the matrix that contains all the aggregated feature vectors of the nodes in network state G_t, $\mathcal{A}_G$ is the adjacency matrix that describes the physical connections in the topology of the EO-DCI $G(V, E)$, $f(\cdot)$ is the function to map the feature vectors of the nodes in G_t to a hidden state H_t, W denotes the learnable parameters of the GNN, and $\sigma(\cdot)$ is an activation function. Note that, since the spectrum usages on the fiber links in the EO-DCI should also be considered here, we abstract the information about them in the adjacency matrix $\mathcal{A}_G$ too. The details about this will be explained in Section V-B.

For the encoder to analyze the context of the provisioning scheme of $\mathcal{R}^t$, it considers the current vNF (δ_k^t) and the node selected for the previous vNF (a_t^{k-1}), which is the "current" node in the vNF-SC provisioning process. The current vNF and the current node are represented as two vectors, which are concatenated into a feature vector. The feature vector is then processed by a long-short-term memory (LSTM), as shown in Fig. 3. Here, the LSTM is used to memorize the determined part of the provisioning scheme of $\mathcal{R}^t$, since the placement of the vNFs before δ_k^t has a great influence on its node selection. The hidden state from the LSTM is a query vector q_t, which is sent to the decoder for selecting the node for δ_k^t.

2) Decoder: It selects a node for the current vNF δ_k^t based on the outputs of the encoders, *i.e.*, H_t and q_t. We design the decoder based on the attention mechanism [46], which can address the problem of long sequence information loss when handling sequential tasks. Specifically, it calculates the weights of all the nodes in the EO-DCI, based on H_t and q_t, and for a node $v \in V$, its weight is defined as

$$u_v^k = W_a \cdot \tanh(W_r \cdot r_v + W_q \cdot q_t), \tag{5}$$

where u_v^k is the weight for node v (k is for the index of δ_k^t), W_a, W_r and W_q are the learnable parameters, r_v is the reference context vector in the hidden state H_t for node v. The function of $\tanh(\cdot)$ has the following expression

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{6}$$

which can re-scale the real elements of a vector x within $[-1, 1]$. Then, the probability of selecting node v for δ_k^t is

$$p_v^k = \text{softmax}(u_v^k), \tag{7}$$

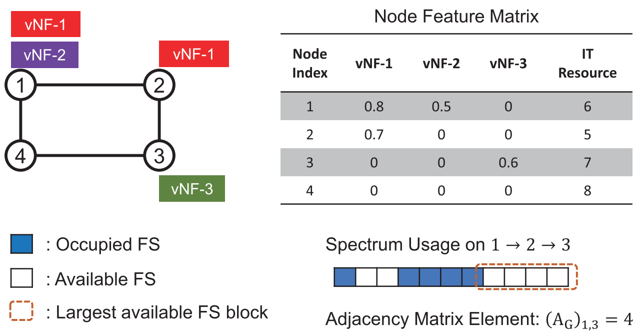
where $\text{softmax}(\cdot)$ is defined as

$$\text{softmax}(x[j]) = \frac{e^{x[j]}}{\sum_{k=1}^{|x|}(e^{x[k]})}, \quad j \in [1, |x|], \tag{8}$$

which can normalize the j-th element $x[j]$ of a real vector x within $[0, 1]$ such that $\sum_{j=1}^{|x|} \text{softmax}(x[j]) = 1$ ($|x|$ denotes the number of elements in x), and the normalized value can be used as the probability of selecting $x[j]$ from x. Then, we can sample the nodes in V according to $\{p_v^k\}$ or choose the one with the largest probability, to find the node for δ_k^t.

B. GNN in Encoder

The GNN learns the high-dimensional representation of the nodes in the graph-structured network state $G_t(V, E)$, which can help the decoder to accomplish node selection. In network state G_t, the feature vector of each node $v \in V$ is a vector x^v, which includes $M + 1$ dimensions. Here, each dimension of the feature vector denotes either the remaining process capability of the existing type-m vNFs ($m \in [1, M]$) or the available IT resources, on node v. The feature vectors of all the nodes in V can be combined into a matrix X_V, whose dimension is $[(M + 1) \times |V|]$. Fig. 4 shows an example on how to calculate X_V based on the network state of an EO-DCI. As there are 4 nodes in the EO-DCI and 3 types of vNFs can be supported, the dimension



vNF-1
vNF-2
vNF-1
1
2
4
3
vNF-3
Node Feature Matrix

Node Index	vNF-1	vNF-2	vNF-3	IT Resource
1	0.8	0.5	0	6
2	0.7	0	0	5
3	0	0	0.6	7
4	0	0	0	8

🟦 : Occupied FS
☐ : Available FS
⬚ : Largest available FS block
Spectrum Usage on 1 → 2 → 3
Adjacency Matrix Element: $(A_G)_{1,3} = 4$

Fig. 4. Example on obtaining matrices for node features and adjacency.

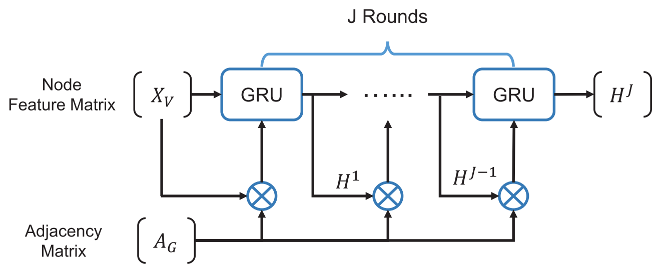


Fig. 5. Structure of our proposed GNN.

of X_V is $[4 \times 4]$. We can also see in Fig. 4 that if a type of vNFs does not exist on node v, the corresponding element in the node's feature vector x^v is set as 0.

Eq. (4) indicates that in addition to X_V, the adjacency matrix $\mathcal{A}_G$ is also needed to calculate the graph-based hidden state H_t. Note that, different from the adjacency matrix of a packet network, $\mathcal{A}_G$ should record the availability of frequency slots (FS') between each node pair in the EO-DCI, because this information is essential for calculating the RSA schemes of lightpaths. Therefore, we define the element in the adjacency matrix $\mathcal{A}_G$ for node pair v_1 and v_2 ($(\mathcal{A}_G)_{v_1,v_2}$) as the size of the largest available FS block on the fiber links on the shortest path between v_1 and v_2. For instance, in Fig. 4, the spectrum usage on the fiber links between *Nodes* 1 and 3 is shown in the lower right corner, which suggests that the size of the largest available FS block is 4, and thus, we have $(\mathcal{A}_G)_{1,3} = 4$.

To process X_V and $\mathcal{A}_G$, we design the GNN based on the gated graph sequence neural network (GGS-NN) [47], which is known to be effective on extracting features about a long node sequence in graph-structured data. The detailed structure and operation principle of the GNN is shown in Fig. 5. To model the long sequence information and get stable node representation, we repeat the information propagation process in the GNN for J iterations. In each iteration, we have both information passing and information aggregation. The information passing is realized by

$$\varpi_v^{j+1} = \sum_{v' \in \text{adj}(v)} \text{relu}(W \cdot h_{v'}^j), \quad \forall v \in V. \tag{9}$$

where W still denotes the learnable parameters of the GNN (as in (4)) and it is shared across J iterations, $h_{v'}^j$ is the hidden state of node v' at round j, $\text{adj}(v)$ returns all the neighbor nodes of a node v, and $\mathrm{relu}(\cdot)$ is defined as

$$\mathrm{relu}(x) = \max(0, x), \tag{10}$$

which has been widely used in DNNs to avoid the gradient vanishing problem. In the information aggregation, we use a gated recurrent unit (GRU) to update the representation of each node. Here, GRU is a variant of LSTM, which combines the forget gate and the input gate of LSTM as an update gate. Hence, GRU has fewer parameters than LSTM and thus it converges faster in training. The detailed operations for the information aggregation in the j-th iteration are as follows.

$$\begin{cases} p_v^{j+1} = \mathrm{softmax}\left(\mathcal{W}_j^p \cdot \varpi_v^{j+1} + \mathcal{U}_j^p \cdot h_v^j\right), \\ q_v^{j+1} = \mathrm{softmax}\left(\mathcal{W}_j^q \cdot \varpi_v^{j+1} + \mathcal{U}_j^q \cdot h_v^j\right), \end{cases} \tag{11}$$

$$\tilde{h}_v^{j+1} = \tanh\left[\mathcal{W}_j^m \cdot \varpi_v^{j+1} + \mathcal{U}_j^m \cdot \left(p_v^{j+1} \odot h_v^j\right)\right], \tag{12}$$

$$h_v^{j+1} = (1 - q_v^{j+1}) \odot \tilde{h}_v^{j+1} + q_v^{j+1} \odot h_v^j, \tag{13}$$

where $\{\mathcal{W}_j^m, \mathcal{U}_j^m\}$, $\{\mathcal{W}_j^p, \mathcal{U}_j^p\}$, and $\{\mathcal{W}_j^q, \mathcal{U}_j^q\}$ are the learnable parameters of the memory cell, update gate and reset gate of the GRU. After we repeating the information passing and aggregation with (9)–(13) for J times, the hidden state of all the nodes in G_t can be obtained as $H_t = \{h_v^J, \forall v \in V\}$.

C. Optimization of GNN-Based Policy Neural Network

All the learnable parameters of the GNN are optimized in the hierarchical training discussed in Section IV-B. Specifically, for both the lower-level and upper-level training processes, the parameters are optimized with the policy gradient method. With (1), the policy gradient in the lower-level training is

$$\nabla_{\theta_\pi} J^{\text{lower}} = \frac{1}{N}\sum_{i=1}^{N}\left\{\left(\sum_{k=1}^{K^t} r_t^{k,i} - \vartheta^i\right) \cdot \left[\sum_{k=1}^{K^t} \nabla_{\theta_\pi} \log\left(\pi\left(a_t^{k,i} | G_t, \delta_{k,i}^t\right)\right)\right]\right\}, \tag{14}$$

where N is the batch size of training samples, $r_t^{k,i}$ is the reward of action $a_t^{k,i}$ in the i-th batch, ϑ^i is the baseline for learning. In order to make the training more stable, the baseline ϑ^i is introduced to calculate the benefit of each action as

$$\vartheta^i = \left(\sum_{k=1}^{K^t} \hat{r}_t^{k,i}\right) + \left[\frac{1}{N}\sum_{k=1}^{K^t}\sum_{j=1}^{N}(r_t^{k,j} - \hat{r}_t^{k,j})\right], \tag{15}$$

where the reward $\hat{r}_t^{k,j}$ is obtained by applying the greedy scheme that chooses the action with the largest probability, and the reward $r_t^{k,j}$ is got with the sampling scheme that samples the actions according to their probabilities. Hence, the second term in the (15) denotes the gap between the rewards from the sampling and greedy schemes. Meanwhile, the policy gradient in the upper-level training is

$$\nabla_{\theta_\pi} J^{\text{upper}} = \sum_{j=1}^{M} r_{t+j} \cdot \nabla_{\theta_\pi} \log\left[\pi(a_t | G_t)\right]. \tag{16}$$

Algorithm 1: GNN Optimization in Lower-level Training.

```
1  initialize parameters $\theta_\pi$ of the GNN;
2  empty the lower-level experience buffer;
3  for each newly-arrived vNF-SC request $\mathcal{R}^t$ do
4      release the resources used by expired vNF-SCs;
5      get current graph-structured state $G_t$ of the EO-DCI;
6      for $k \in [1, K^t]$ do
7          use the GNN to get the node selection policy
           $\pi(|G_t, \delta_k^t)$ for vNF $\delta_k^t$;
8          sample the nodes in $V$ according to $\pi(|G_t, \delta_k^t)$ to
           get a proper action $a_t^k$;
9          store $<(G_t, \delta_k^t), a_t^k, r_t^k, (G_t, \delta_{k+1}^t)>$ as a training
           sample in lower-level experience buffer;
10     end
11 end
12 select the training samples about $N$ vNF-SCs from
   lower-level experience buffer randomly;
13 calculate the baseline $\{\vartheta^i,\ i \in [1, N]\}$ with Eq. (15);
14 get $J^{\text{lower}}(\theta_\pi)$ and $\nabla_{\theta_\pi} J^{\text{lower}}(\theta_\pi)$ with Eqs. (1) and (14);
15 update $\theta_\pi$ with the stochastic gradient descent method;
```

Algorithm 2: GNN Optimization in Upper-level Training.

```
1  empty the upper-level experience buffer;
2  for each newly-arrived vNF-SC request $\mathcal{R}^t$ do
3      release the resources used by expired vNF-SCs;
4      get current graph-structured state $G_t$ of the EO-DCI;
5      provision $\mathcal{R}^t$ with the GNN;
6      store $<(G_t, \mathcal{R}^t), a_t, r_t, (G_{t+1}, \mathcal{R}^{t+1})>$ as a training
       sample in upper-level experience buffer;
7      if the number of samples exceeds $M$ then
8          get $J^{\text{upper}}$ and $\nabla_{\theta_\pi} J^{\text{upper}}(\theta_\pi)$ with Eqs. (2) and
           (16), respectively;
9          update $\theta_\pi$ with stochastic gradient descent;
10     end
11 end
```

The detailed procedure of optimizing the parameters of the GNN in the lower-level training is explained in *Algorithm* V-C. *Lines* 1 and 2 are for the initialization. Then, for each newly-arrived request R^t, we determine its provisioning scheme with our hierarchical DRL and record the result as a training sample in the lower-level experience buffer (*Lines* 3–11). *Line* 12 randomly selects a batch of N vNF-SCs from the lower-level experience buffer, which are used to optimize the parameters of the GNN in *Lines* 13–15. Specifically, we calculate the objective of the lower-level training and its gradient with (1) and (14) in *Line* 14, and update the parameters of the GNN with the stochastic gradient descent method in *Line* 15.

After the lower-level training has converged, we trigger the upper-level training to further optimize the parameters of the GNN, such that it can make smart decisions on vNF-SC provisioning to reduce the overall blocking probability. We determine that the lower-level training has converged if in the last 100 training steps, the maximum change on its reward is less than 1%. The procedure is described in *Algorithm* V-C. *Line* 1 is

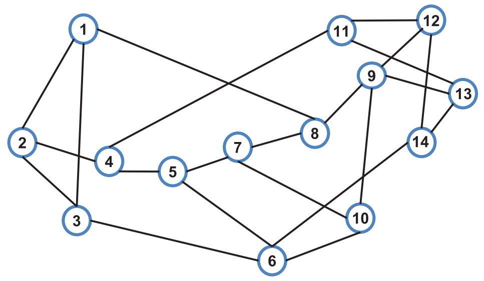


Fig. 6. NSFNET topology.

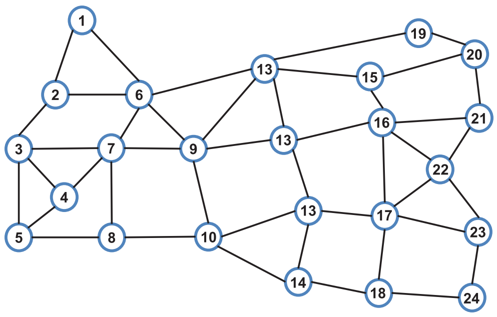


Fig. 7. USB topology.

for the initialization. Then, the for-loop that covers *Lines* 2–11 provisions each vNF-SC with our hierarchical DRL. This time, we record the whole provisioning scheme of each vNF-SC request as a training sample in the upper-level experience buffer (*Line* 6), and when enough training samples have been accumulated, an upper-level training is triggered (*Lines* 7–10).

VI. Performance Evaluation

In this section, we discuss the numerical simulations to evaluate the performance of our proposal.

A. Simulation Setup

Our simulations consider two EO-DCI topologies, *i.e.*, the 14-node NSFNET and the 24-node US backbone (USB) topologies shown in Figs. 6 and 7, respectively. We assume that each fiber link[3] can accommodate $F = 358$ FS,' and each DC node possesses $C_v = 100$ units of IT resources for vNF deployment. The EO-DCIs can support $M = 5$ types of vNFs, and an instance of each type of vNFs consumes $[4, 8]$ units of IT resources. Each deployed vNF can be shared by 3 vNF-SC requests at most. Note that, the traffic pattern in a DCI usually follows Poisson traffic model [48], and the same model and its variations have been widely used to describe the traffic patterns of many emerging network services [49]. Hence, our simulations assume that the vNF-SC requests arrive dynamically according to Poisson traffic model. Each of them demands for $[2, 4]$ vNFs and selects the source and destination nodes randomly in the EO-DCI, and its bandwidth requirement is evenly distributed within $[20, 100]$ Gb/s (*i.e.*, $[3, 9]$ FS'). As for the coefficients in (3) for the

[3]We assume that the optical layer of the EO-DCI uses the C-band, *i.e.*, each fiber link has ∼4.475 THz bandwidth to allocate.

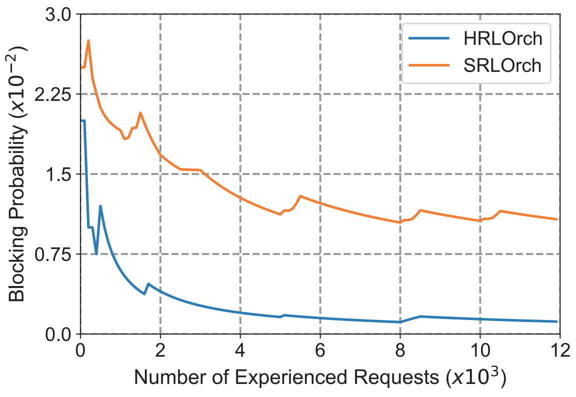


Fig. 8. Evolution of blocking probability from HRLOrch and SRLOrch (traffic load at 700 Erlangs in NSFNET).

instant reward, we empirically set them as $\alpha = 10$, $\beta = 1$, and $\gamma = 1$. The GNN-based policy NN contains 128 hidden layers. For the training of HRLOrch, we select the batch sizes for the lower-level and the upper-level training (N and M) as 8 and 20, respectively. To ensure the statistical accuracy of our results, we conduct 10 independent runs, record the results that are within the 95% confidence interval, and average them to get each data point.

B. Training Performance

We first evaluate the training performance of our proposed HRLOrch in an EO-DCI with the NSFNET topology. As mentioned in Section IV-B, its training process contains a low-level training followed by an upper-level training. Hence, to verify the effectiveness of the hierarchical training scheme, we design a benchmark that is based on a DRL agent using single-level training, namely, SRLOrch. Specifically, SRLOrch uses the same policy NN as that of HRLOrch, but its training process is single-level, which means that the training tries to minimize the blocking probability of vNF-SC requests directly.

Fig. 8 compares the evolution of the blocking probability from HRLOrch and SRLOrch in their training processes. We can see that the blocking probability from HRLOrch quickly converges to 1.1×10^{-3} after experiencing $\sim 6 \times 10^3$ vNF-SC requests (*i.e.*, running for ∼2,160 seconds), while SRLOrch cannot provide a stable blocking probability even after experiencing 1.2×10^4 requests (*i.e.*, running for ∼4,190 seconds), and its blocking probability is much higher than that from HRLOrch. This is because SRLOrch trains the policy NN to learn how to minimize the blocking probability directly, which can hardly be effective due to the fact that the EO-DCI is a sparse reward environment for vNF-SC provisioning. To this end, the results in Fig. 8 confirm that our proposed hierarchical training scheme can effectively address the sparse reward environment for vNF-SC provisioning, and obtain better training performance on blocking probability reduction.

We hope to point out that the training is not completely independent of traffic load. Specifically, if we want to apply the HRLOrch that has been trained for one traffic load to an EO-DCI whose traffic load is different, its parameters need to be

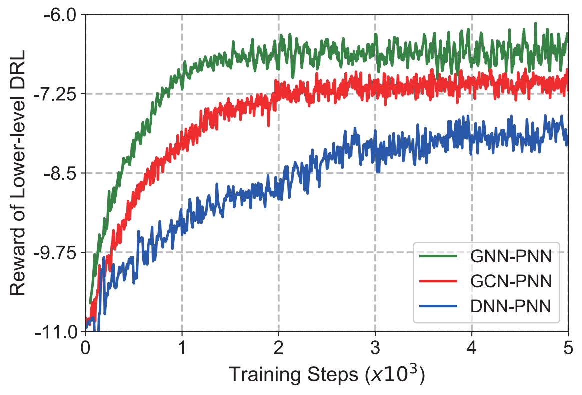


Fig. 9. Evolution of reward from HRLOrch with different policy NNs.

fine-tuned with transfer learning [50]. In other words, we do not need to retrain HRLOrch for each traffic load from the scratch. We actually conduct simulations with various traffic loads to verify this, and confirm that HRLOrch can always outperform SRLOrch with a similar trend as that in Fig. 8.

Next, we would like to justify our design of the GNN-based policy NN (GNN-PNN) in HRLOrch. Therefore, we design two benchmark policy NNs based on the classical architectures, *i.e.*, the graph convolution network (GCN) [51] and deep neural network (DNN) [46] based policy NNs (namely, GCN-PNN and DNN-PNN, respectively), which are known to have relatively good performance on policy selection [46], [51]. GCN-PNN and DNN-PNN use the same encoder-decoder structure as our proposed GNN-PNN, but their NNs for network state extraction are different. Specifically, GCN-PNN uses a two-layer GCN to extract the graph-structured network state directly, while DNN-PNN concatenates the feature vectors of all the nodes and links into a vector and inputs it to a DNN to map to a hidden network state.

The training performance of the three policy NNs in the lower-level training is plotted in Fig. 9, which shows the evolution of the reward defined in (3) in the training. We observe that our proposed GNN-PNN provides the best convergence performance among the three policy NNs. As GNN-PNN has less learnable parameters than DNN-PNN and it can operate on graph-structured data directly, it converges much faster. Meanwhile, since GNN-PNN can extract the graph-structured network state more precisely than GCN-PNN, it can achieve a larger reward after convergence.

C. Performance on Dynamic vNF-SCs Provisioning

We then evaluate the performance of HRLOrch for the dynamic provisioning of vNF-SCs in an EO-DCI. Here, we still assume that the EO-DCI uses the NSFNET topology. Three existing heuristics for vNF-SC provisioning in EO-DCIs are considered as benchmarks, which are the greedy placement of vNFs and shortest-path routing (GP-SPR), maximizing vNF reuses and spectrum-saving routing (MRP-SSR), and balanced placement of vNFs and shortest-path routing (BP-SPR) [19]. Specifically, the benchmark algorithms work as follows. GP-SPR tries to greedily place vNFs along the shortest path from the source to the destination of each vNF-SC. MP-SSR first tries to save IT resources by reusing the existing vNFs in the EO-DCI, and then deploys new vNFs in the way that the spectrum usages on fiber links can be minimized. BP-SPR places vNFs along the shortest path that can balance the IT and spectrum resource usages in the EO-DCI. Fig. 10 shows the performance comparisons of dynamic vNF-SC provisioning. Fig. 10(a) shows that our proposed HRLOrch provides the lowest blocking probability at all the traffic loads, which verifies the performance of HRLOrch on dynamic vNF-SC provisioning. Moreover, it is interesting to observe that our proposed HRLOrch has the highest IT and spectrum resource usages than the benchmark algorithms in Figs. 10(b) and 10(c). This further verifies the effectiveness of HRLOrch, *i.e.*, it can leverage the hierarchical training scheme to achieve the best utilization of the IT and spectrum resources in the EO-DCI.

D. Evaluations on Universality and Scalability

Finally, we investigate the universality and scalability of HRLOrch. Firstly, for its universality, we consider a zero-shot transfer learning scenario (zero-learning) [52], which refers to applying a trained DRL model to an unseen environment for the same task. We apply the HRLOrch that has been trained in the EO-DCI with the NSFNET topology to one with the USB topology for the vNF-SC provisioning in it. The USB topology is shown in Fig. 7, and we keep the settings of the simulation parameters about IT and spectrum resources as unchanged.

The procedure of the zero-learning can be seen in Fig. 11, where we fix the traffic load as 1,000 Erlangs in USB. We observe that for this case, the HRLOrch trained in NSFNET provides lower blocking probability than GP-SPR in USB even without the zero-learning, and its blocking probability quickly goes below that of MRP-SSR and BP-SPR after being retrained with only 500 vNF-SC requests in USB. This confirms the universality of HRLOrch. Meanwhile, in Fig. 11, we also plot the training performance of scratch-learning of HRLOrch, which refers to the training scheme that starts the training of HRLOrch from the scratch in USB. We observe that the zero-learning scheme provides lower blocking probability and converges faster than the scratch-learning scheme. This confirms that the GNN-based policy NN in HRLOrch can extract high-level and universal knowledge about vNF-SC provisioning in an EO-DCI, and the learned knowledge is still useful even after the EO-DCI changes its topology.

Fig. 12 shows the blocking probability from the algorithms in USB, and with a similar trend as that in Fig. 10(a), HRLOrch still outperforms all the benchmarks significantly. These results verify that HRLOrch can adapt to an arbitrary EO-DCI topology without changing its architecture, *i.e.*, the GNN-PNN in HRLOrch can operate on the graph-structured network state of an EO-DCI directly and effectively.

Secondly, regarding the scalability of HRLOrch, we summarize the average running time for the algorithms to serve a vNF-SC request in Table I. It can be seen that due to its online training, the running time of HRLOrch is longer than that of the benchmarks, but the results are still comparable and only

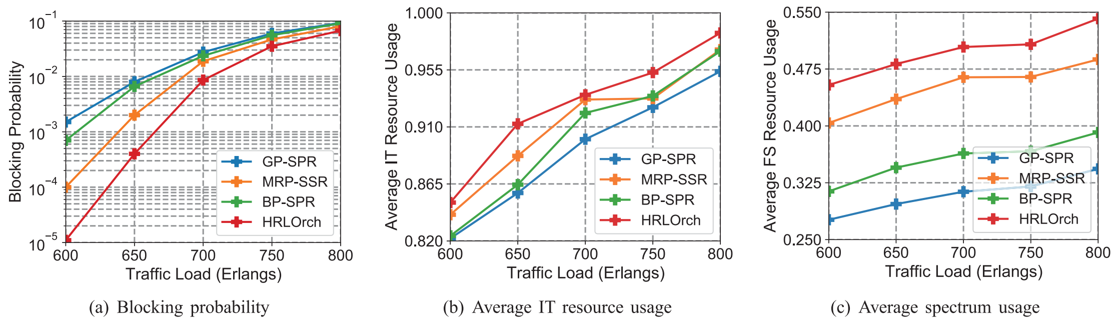


Fig. 10. Performance comparisons of dynamic vNF-SCs provisioning under different traffic loads (NSFNET topology).

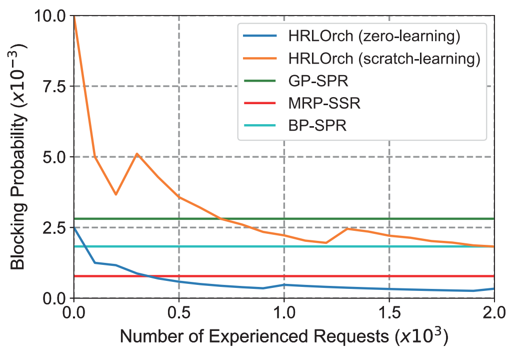


Fig. 11. Evolution of blocking probability from HRLOrch (traffic load at 1,000 Erlangs in USB).

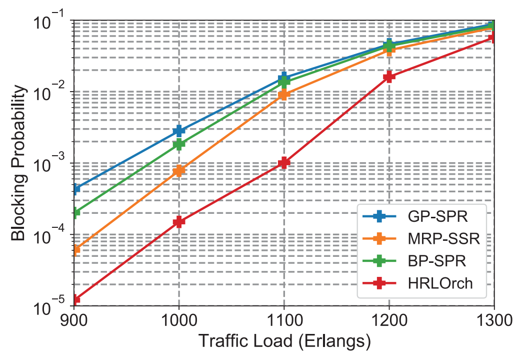


Fig. 12. Results on blocking probability in USB.

TABLE I
RUNNING TIME PER vNF-SC REQUEST

Algorithm	Running Time (msec)	
	EO-DCI with NSFNET	EO-DCI with USB
GP-SPR	12.06	30.81
MP-SSR	19.47	32.16
BP-SPR	13.78	33.95
HRLOrch	37.74	49.76

in tens of milliseconds. This confirms that HRLOrch is suitable for online and dynamic vNF-SC provisioning. Meanwhile, it is interesting to notice that when the size of the EO-DCI increases (*i.e.*, from 14-node NSFNET to 24-node USB), the running time increases for 2.55, 1.65 and 2.46 times for GP-SPR, MP-SSR and BP-SPR, respectively, but that of HRLOrch only increases 1.32 times. This suggests that HRLOrch has better scalability than the benchmarks, which is because the structure of the GNN-PNN in HRLOrch does not need to be changed when the EO-DCI topology changes.

VII. Conclusion

In this work, we proposed a GNN-based hierarchial DRL model, namely, HRLOrch, for realizing online vNF-SCs provisioning in an EO-DCI. To ensure the universality and scalability of HRLOrch, we designed the policy NN in it based on a GNN, which can operate on the graph-structured network state of the EO-DCI directly and can adapt to an arbitrary EO-DCI topology without any structural changes. Then, to address the issue that the EO-DCI is a sparse reward environment for vNF-SC provisioning, we introduced a hierarchical DRL model, which divides the traditional DRL into lower-level and upper-level models, assigns different optimization objectives to them, and makes them operate cooperatively in the training to minimize the blocking probability of vNF-SC requests.

Our simulation results demonstrated that the proposed hierarchical DRL model achieved better convergence performance than the traditional single-level DRL model in training. Moreover, due to its intelligence, HRLOrch provided lower blocking probability than the existing heuristics for vNF-SC provisioning in an EO-DCI. Finally, the simulations also confirmed the universality and scalability of HRLOrch, because the HRLOrch trained in one EO-DCI topology could adapt to a new EO-DCI topology quickly in zero-shot transfer learning, still provide lower blocking probability than the existing heuristics, and its running time increased slower than the heuristics when the size of the EO-DCI became larger.

References

[1] *Cisco visual networking index: Forecast and methodology*, 2017–2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html#_Toc529314186
[2] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, no. 5, pp. 36–42, Sep./Oct. 2015.
[3] "Network functions virtualization (NFV)," Tech. Rep., Oct. 2014. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf
[4] "Network functions virtualisation (NFV): Use cases," Tech. Rep., Oct. 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf

[5] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surv. Tut.*, vol. 18, no. 1, pp. 236–262, Jan.–Mar. 2016.

[6] Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement," in *Proc. IEEE Glob. Commun. Conf.*, 2016, pp. 1–6.

[7] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, pp. 3330–3341, Jul. 2016.

[8] Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," *J. Lightw. Technol.*, vol. 35, pp. 2712–2723, Jul. 2017.

[9] K. Han *et al.*, "Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing," *IEEE Access*, vol. 6, pp. 26567–26577, 2018.

[10] L. Dong *et al.*, "On application-aware and on-demand service composition in heterogenous NFV environments," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.

[11] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in Geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, pp. 3005–3015, Jul. 2015.

[12] X. Xie, Q. Ling, P. Lu, W. Xu, and Z. Zhu, "Evacuate before too late: Distributed backup in inter-DC networks with progressive disasters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 5, pp. 1058–1074, May 2018.

[13] W. Fang *et al.*, "Joint defragmentation of optical spectrum and IT resources in elastic optical datacenter interconnections," *J. Opt. Commun. Netw.*, vol. 7, pp. 314–324, Mar. 2015.

[14] Z. Pan *et al.*, "Advanced optical-label routing system supporting multicast, optical TTL, and multimedia applications," *J. Lightw. Technol.*, vol. 23, pp. 3270–3281, Oct. 2005.

[15] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[16] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[17] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[18] X. Chen, F. Ji, and Z. Zhu, "Service availability oriented p-cycle protection design in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 6, pp. 901–910, Oct. 2014.

[19] W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and IT resource allocation for efficient vNF service chaining in inter-datacenter elastic optical networks," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1539–1542, Aug. 2016.

[20] X. Chen *et al.*, "Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks," *J. Opt. Commun. Netw.*, vol. 10, pp. A232–A240, Feb. 2018.

[21] W. Lu, L. Liang, and Z. Zhu, "Orchestrating data-intensive VNF service chains in inter-DC elastic optical networks," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, 2017, pp. 1–6.

[22] X. Chen, Z. Zhu, R. Proietti, and B. Yoo, "On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: A hierarchical game-theoretic mechanism," *IEEE Trans. Netw. Serv. Manage.*, vol. 16, no. 1, pp. 1–12, Mar. 2019.

[23] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/first/the-book.html

[24] X. Chen *et al.*, "Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks," *J. Lightw. Technol.*, vol. 37, pp. 1742–1749, Apr. 2019.

[25] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 263–278, Feb. 2020.

[26] X. Chen *et al.*, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," *J. Lightw. Technol.*, vol. 37, pp. 4155–4163, Aug. 2019.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[28] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 1–9.

[29] M. Chowdhury, M. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[30] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[31] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding (A-SVNE) in optical datacenter networks," *J. Opt. Commun. Netw.*, vol. 7, pp. 1160–1171, Dec. 2015.

[32] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3648–3661, Dec. 2016.

[33] I. Jang, D. Suh, S. Pack, and G. Dan, "Joint optimization of service function placement and flow distribution for service function chaining," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2532–2541, Nov. 2017.

[34] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.

[35] M. Dieye *et al.*, "CPVNF: Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 15, no. 2, pp. 774–786, Jun. 2018.

[36] M. Lopez, D. Mattos, and O. Duarte, "Evaluating allocation heuristics for an efficient virtual network function chaining," in *Proc. 7th Int. Conf. Netw. Future*, 2016, pp. 1–5.

[37] L. Askari, A. Hmaity, F. Musumeci, and M. Tornatore, "Virtual-network-function placement for dynamic service chaining in metro-area networks," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, 2018, pp. 136–141.

[38] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks," *J. Opt. Commun. Netw.*, vol. 10, pp. D29–D41, Oct. 2018.

[39] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in EON-DCIs," *J. Opt. Commun. Netw.*, vol. 12, pp. A18–A27, Jan. 2020.

[40] K. Rusek *et al.*, "Unveiling the potential of graph neural networks for network modeling and optimization in SDN," in *Proc. ACM Symp. SDN Res.*, 2019, pp. 140–151.

[41] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1520–1523, Sep. 2012.

[42] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, pp. 1354–1366, May 2011.

[43] K. Lu, G. Xiao, and I. Chlamtac, "Analysis of blocking probability for distributed lightpath establishment in WDM optical networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 187–197, Mar. 2005.

[44] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications," *IEEE Access*, vol. 7, pp. 802–822, 2018.

[45] T. He *et al.*, "Layer-wise coordination between encoder and decoder for neural machine translation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7955–7965.

[46] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Jan. 2014, *arXiv:1409.0473*.

[47] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," Sep. 2017, *arXiv:1511.05493*.

[48] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "Modeling internet backbone traffic at the flow level," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2111–2124, Aug. 2003.

[49] J. Navarro-Ortiz, P. Romero Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos Munoz, and J. M. Lopez Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surv. Tut.*, vol. 22, no. 2, pp. 905–929, Apr.–Jun. 2020.

[50] M. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, Jul. 2009.

[51] X. Bresson and T. Laurent, "A two-step graph convolutional decoder for molecule generation," 2019. [Online]. Available: http://arxiv.org/abs/1906.03412

[52] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4166–4174.