mathematics

MDPI

Article

Deep Reinforcement Learning-Based RMSA Policy Distillation for Elastic Optical Networks

Bixia Tang [1], Yue-Cai Huang [2,*], Yun Xue [1,2] and Weixing Zhou [1,2]

[1] School of Physics and Telecommunication Engineering, South China Normal University, Guangzhou 510006, China
[2] School of Electronics and Information Engineering, South China Normal University, Foshan 528200, China
* Correspondence: huangyuecai@scnu.edu.cn

Abstract: The reinforcement learning-based routing, modulation, and spectrum assignment has been regarded as an emerging paradigm for resource allocation in the elastic optical networks. One limitation is that the learning process is highly dependent on the training environment, such as the traffic pattern or the optical network topology. Therefore, re-training is required in case of network topology or traffic pattern variations, which consumes a great amount of computation power and time. To ease the requirement of re-training, we propose a policy distillation scheme, which distills knowledge from a well-trained teacher model and then transfers the knowledge to the to-be-trained student model, so that the training of the latter can be accelerated. Specifically, the teacher model is trained for one training environment (e.g., the topology and traffic pattern) and the student model is for another training environment. The simulation results indicate that our proposed method can effectively speed up the training process of the student model, and it even leads to a lower blocking probability, compared with the case that the student model is trained without knowledge distillation.

Keywords: routing, modulation and spectrum assignment; elastic optical networks; deep reinforcement learning; knowledge distillation

MSC: 68T07

Citation: Tang, B.; Huang, Y.-C.; Xue, Y.; Zhou, W. Deep Reinforcement Learning-Based RMSA Policy Distillation for Elastic Optical Networks. *Mathematics* **2022**, *10*, 3293. https://doi.org/10.3390/math10183293

Academic Editors: Jianping Gou, Weihua Ou, Shaoning Zeng and Lan Du

Received: 14 August 2022
Accepted: 8 September 2022
Published: 11 September 2022

1. Introduction

Accompanied with the rapid development of the Internet technology, services such as audio and video conferencing, webcasting, and cloud computing have become popular. The growing demand of these services leads to an exponential increase in data traffic and poses great challenges to the bearing communication networks [1]. Elastic optical networks (EONs) have been regarded to be a promising candidate for the next-generation optical communications [2,3]. In EONs, the spectrum is divided into narrow frequency slots, and traffic requests can be served by different numbers of frequency slots according to their data rate requirements and the quality of the connection. This flex-grid scheme greatly increases the network resource allocation flexibility compared to the traditional wavelength-division multiplexing (WDM)-based networks [4]. Meanwhile, it also brings difficulties for the network resource management.

The routing, modulation, and spectrum assignment (RMSA) [5] is a key problem for the EONs resource management. Due to the complexity, the RMSA problem is generally divided into two sub-problems: the routing and spectrum assignment [6], each of them tackled by heuristic solutions [7–10]. For the routing sub-problem, representative approaches include fixed routing, fixed alternative routing [11,12], and adaptive routing [4]. For the spectrum assignment sub-problem, there are the first-fit [13] and random-fit schemes and other methods. However, these rule-based heuristics, mostly relying on researchers' cognition, cannot comprehensively capture the effect of the complex network conditions.

To overcome the above limitation, deep reinforcement learning (DRL) has recently been introduced to the RMSA problem [14–19], where the RMSA policies are parameterized by deep neural networks and the RMSA policies are improved through interactions with the optical network environment. Many of them have achieved a better performance than heuristic methods. However, the learned policies of these DRL-based approaches are highly related to the training environment, such as the traffic patterns and the network topologies. However, in a practical network, the traffic patterns and the network topologies are very likely to be changed. For example, the traffic volume from commercial and residential areas varies from working hours to off-duty hours. Meanwhile, the network topology becomes different in the case of a network failure or disasters. Once the environment is changed, the effectiveness of the learned RMSA policies deteriorates significantly [20]. Therefore, re-training is required and consumes a lot of computing power and time. To ease the requirement of re-training, Chen et al. [20] investigated the transfer learning (TL) between different network topologies. They first trained and obtained a model from source tasks, and then copied the parameters of the trained model as the starting point when training the target task. The limitation is that the target task needs to use the same neural network architecture with the source task. Moreover, the effect of traffic variation has not yet been investigated.

In this paper, we extend our previously published conference paper [19] and apply policy distillation [21] to the RMSA problem, combining knowledge distillation [22] with reinforcement learning (RL). First, a teacher model is trained for one task with a specific traffic pattern and network topology. Then, the well-trained policy of a teacher model is distilled, and the knowledge is transferred to a student model with a different traffic pattern and network topology, to assist the training of the student model. A major difference between our work and the transfer learning in [20] is that the student model (target) and the teacher model (source) can be different. This allows knowledge transfer in a broader context. We have applied the proposed design in three different application scenarios, which consider different traffic patterns and different topologies. The simulation results demonstrate that policy distillation can accelerate the training speed of the student model and improve its performance.

The rest of this paper is organized as follows. Section 2 surveys the related work. In Section 3, we briefly introduce some basics of RL. In Section 4, we introduce the proposed policy distillation architecture, including the problem formulation and the training of the teacher model and the student model. Then, we present the simulation results in Section 5. Lastly, we conclude the paper in Section 6.

2. Related Work

2.1. Deep Reinforcement Learning in RMSA of EONs

In recent years, research has emerged by exploiting DRL to solve the routing and spectrum assignment problem in the optical networks. Chen et al. [23] proposed a DRL framework, namely DeepRMSA, for the optical network management and resource allocation. The DeepRMSA uses the deep Q-learning algorithm for the training. Because the input-state representation has a significant impact on the performance, a series of work has explored different state representations. Chen et al. [14] defined a list of features of the candidate paths. Yan et al. [24] introduced the concept of a multi-modal optical network by considering the topology modality and routing modality to represent different features of the optical network and uses the actor–critic (AC) algorithm for the training. Suárez-Varela et al. [25] captured the key relationships between the links in the input-state representation, making the DRL agents easier and faster to learn. The same team then [26] introduced the Graph Neural Networks to further capture the network-state features. Xu et al. [18] introduced a link–path relationship matrix to capture the path information of the elastic optical networks.

There are some other works exploring various aspects by applying DRL in the optical network management. Huang et al. [15] proposed a DRL-based self-learning routing

scheme for the WDM-based networks. It allows the agent to continuously improve its performance by self-comparison. Koch et al. [27] adopted the RL algorithm for parameter optimization in EONs. In addition, a cost-efficient routing, modulation, wavelength, and port assignment algorithm based on DRL was developed in [28]. Moreover, Li et al. [29] investigated collaborative DRL agents for multi-domain provisioning in multi-area optical networks.

2.2. Transfer Learning in EONs

Transfer learning in EONs has recently attracted research interest. Yao et al. [30] proposed a TL-based resource optimization strategy for predicting the spectrum defragmentation time in space-division multiplexing EONs. Liu et al. [31] applied a TL approach to implement a scalable quality-of-transmission estimation in EONs. To our knowledge, the most relevant work of this paper is [20], where the authors propose a knowledge transfer design that alleviates scalability issues by transferring knowledge between RMSA agents with different tasks through a modular DRL agent structure. As mentioned in Section 1, its limitation is that the target task needs to use the same neural network architecture with the source task. In our previously published conference paper [19], we propose a knowledge distillation scheme based on DRL to achieve RMSA policy scalability in EONs. This paper extends [19] in three aspects: (1) the authors of [19] only consider different traffic patterns, while this paper considers different traffic patterns and topologies; (2) the training algorithm is updated to the most advanced asynchronous advantage actor–critic (A3C); and (3) many more simulation results are provided to verify our proposal.

3. Preliminaries

As this work is based on RL, we first explain some basics about RL for the facility of the readers.

3.1. Reinforcement Learning

Reinforcement learning is an important branch of machine learning. Many RL tasks can be modeled as Markov decision processes (MDP), expressed as tuples $\{S, A, R, P\}$. S is the state space of the environment; A is the action space of the agent; R is the reward function; and P represents the state transition probabilities. In the RL framework, the agent interacts with the environment. Specifically, given a state $s_t \in S$, the agent performs an action $a_t \in A$ according to a *policy*, and then the environment emits a reward r_t and changes its state from s_t to a new state s_{t+1} according to the state transition probabilities P. In this process, the agent influences the environment by taking the actions, and the environment feeds back reward r_t to the agent, which will guide the agent to choose better actions. The goal of the agent is to improve its action policy by optimizing the cumulative future reward.

3.2. Asynchronous Advantage Actor–Critic

The RL agent needs to be trained by some training algorithm. In this work, we use the A3C algorithm [32] for the training. It is the asynchronous multi-threaded version of the AC algorithm [33]. The AC algorithm uses a policy network (also called actor) to select the action and a value network (also called critic) to evaluate actions. The actor updates its policy (i.e., action selection probability) according to the critic. Through the agent–environment interaction, the critic improves its evaluation accuracy, and the actor improves its policy gradually.

A3C makes the AC algorithm much easier and faster to converge. It adopts a multi-threaded method, where each thread has an independent actor–critic pair interacting with a copy of the environment. Each thread collects the exploration experience from its environment copy and then regularly updates a shared global actor–critic pair. By doing this, the algorithm converges faster.

4. Policy Distillation Design with EONs

4.1. Elastic Optical Networks

In the EONs, the RMSA problem is to establish corresponding end-to-end paths and allocate appropriate frequency slots (FSs) for different traffic requests according to their data rate requirements. Furthermore, RMSA [6] algorithm must satisfy the spectrum contiguity constraint and spectrum continuity constraint. The topology of the EONs can be denoted by a graph $G(V, E)$, where V and E represent the set of nodes and links, respectively. When a traffic request, denoted by $TR(v_s, v_d, b)$, arrives, RMSA is needed from the source node $v_s \in V$ to the target node $v_d \in V$ with the required bandwidth b. The routing algorithm first calculates all possible paths from the source to the destination, then selects one path P_{v_s,v_d} from the K-shortest paths. Corresponding number of FS n required on the selected path P_{v_s,v_d} can be calculated by Equation (1) and Table 1.

$$n = \lceil b/(W \cdot m(P_{v_s,v_d})) \rceil + 1 \tag{1}$$

W denotes the spectrum width of each FS; $m(P_{v_s,v_d}) \in [1, 2, 3, 4]$ corresponds to the modulation format selected according to the physical length of P_{v_s,v_d} [34]; and one FS is used for the guard band. Then, n allocated FSs must be contiguous (spectrum contiguity constraint), and each link along the demand path P_{v_s,v_d} must be assigned the same n contiguous FSs (spectrum continuity constraint).

Table 1. Transmission reach for different modulation formats [35].

$m(P_{v_s,v_d})$	Modulation Format	Transmission Reach
1	BPSK	5000 km
2	QPSK	2500 km
3	8-QAM	1250 km
4	16-QAM	625 km

4.2. Policy Distillation Scheme

We propose to integrate policy distillation into the RMSA problems of the optical networks. The whole architecture is shown in Figure 1. Two models, namely the teacher model and the student model, are trained for different tasks. First, a teacher model is trained for one task with specific traffic pattern and network topology. Then, the well-trained policy of the teacher model is distilled, and the knowledge is transferred to a student model with a different traffic pattern and network topology, to assist the training of the student model. There are three steps in the training process:

- Step 1: Train the teacher model. It is trained by interacting with the teacher environment.
- Step 2: Distill the knowledge from the teacher model and transfer the knowledge to the student model. The training data of the student model are generated by calling the well-trained teacher model obtained in Step 1, and then the student model is trained by fitting these data.
- Step 3: Train the student model by itself. After the training in Step 2, the student model will be further updated by interacting with student environment and no longer rely on the knowledge distilled from the teacher model.

The RMSA policy for the student task is learned by the student model via Steps 2 and 3. Step 2 distills the knowledge from the well-trained policy network of the teacher model and transfers the knowledge to the student model to assist its training.

4.3. State, Action, and Reward

The optical network RMSA problem can be modeled as an MDP and solved in an RL-based framework. In the RL framework, three essential elements are the state, the action, and the reward. We consider the state only when there is a new traffic request. The state s_t is a $1 \times 5K$ vector containing spectrum utilization information on the K-shortest

candidate paths of the traffic request [14]. For each candidate path, we considered five elements of spectrum utilization as follows:

- Starting index of the first available FS-block;
- Size of the first available FS-block;
- Number of required FSs;
- Average size of the available FS-block;
- Total number of available FSs.

In addition, the action of the RMSA problem is to choose one path from the K-candidate paths and allocate spectrum on the selected path based on the first-fit strategy. Therefore, action $a_t \in \{1, 2, \cdots, K\}$. The reward r_t is defined to be 1 when the traffic request is accepted, and -1 otherwise.

4.4. Teacher Model

According to Step 1 in Figure 1, a teacher model is first trained, which is illustrated in more detail in Step 1 of Figure 2. We use DRL to train the teacher model and obtain the RMSA policy to optimize the EONs resource management. The A3C algorithm is adopted for the training, where multiple local actor–critic pairs are trained by interacting with the copies of the environment in parallel, and then periodically update the global actor–critic pair. The actor and critic are parameterized by two neural networks: the policy network $\pi(a_t|s_t;\theta_{p,\mathbb{T}})$ and the value network $V(s_t;\theta_{v,\mathbb{T}})$. The policy network $\pi(a_t|s_t;\theta_{p,\mathbb{T}})$ is used to generate the policy of RMSA, which is represented by a probability distribution. The value network $V(s_t;\theta_{v,\mathbb{T}})$ is used to obtain the value of s_t and evaluate the RMSA policy. $\mathbb{T}$ denotes the teacher model. $\theta_{p,\mathbb{T}}$ and $\theta_{v,\mathbb{T}}$ are the parameters of the policy and the value network, respectively. The global parameters maintained by the A3C algorithm are represented as $\theta^*_{p,\mathbb{T}}$ and $\theta^*_{v,\mathbb{T}}$.

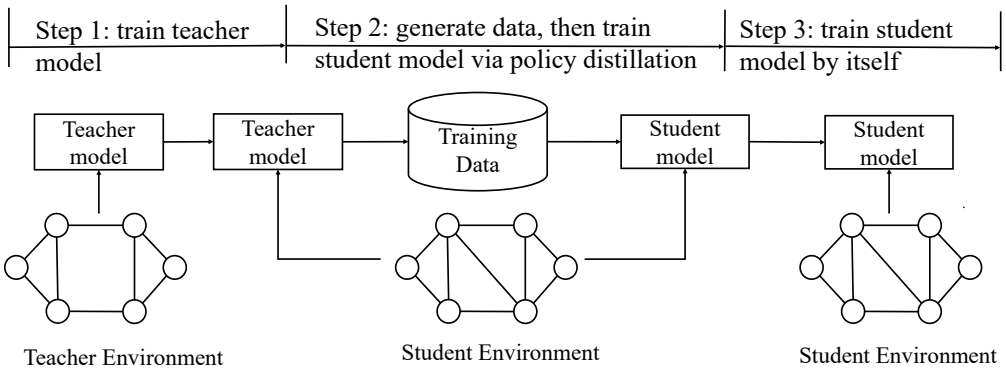


Figure 1. Overview of the policy distillation design with EONs.

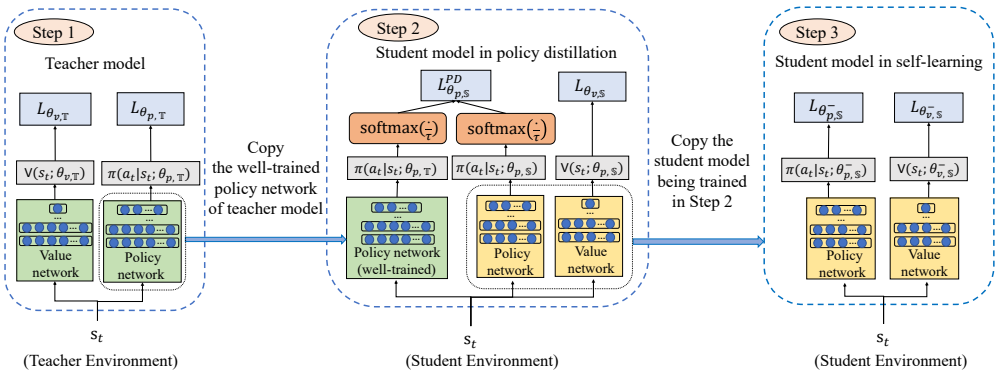


Figure 2. Detailed illustration of policy distillation design with EONs.

The details of training process for the teacher model are given in Algorithm 1. First, we initialize the experience buffer D to empty and set the initial exploration rate ε to 1. In line 3, each actor–critic pair thread parameters are firstly updated by the global parameters. Notice that for a general DRL task that can be modeled as a Markov decision process $\{S, A, R, P\}$ mentioned in Section 3, the state transition from s_t to s_{t+1} follows a probability distribution P. However, for the RMSA task in this paper, as the state space is extremely large, state transitions are difficult to be modeled. Therefore, the RMSA task here belongs to the model-free MDP and can only be optimized through samples. In lines 6–10, during the sampling, we first input the $1 \times 5K$-dimensional state s_t into the policy and value networks. Then, the policy network outputs a $1 \times K$-dimensional probability distribution $\pi(a_t|s_t;\theta_{p,\mathbb{T}})$, where each probability ranges from 0 to 1, and the summation of the output K probabilities is 1. The value network outputs a value $V(s_t;\theta_{v,\mathbb{T}})$, which is a real number. Finally, we store the sample $(s_t, a_t, r_t, V(s_t;\theta_{v,\mathbb{T}}))$ generated by the interaction of the agent and the environment in an experience buffer D. When the size of experience buffer reaches $2N-1$, we perform training based on the first N samples (lines 13–19). For each sample at time t, the advantage function is calculated in line 15. To obtain the advantage function, we first make cumulative the discounted reward for this sample (we only consider an episode consisting of N consecutive samples after this sample and ignore the discounted reward after N samples) by,

$$Q_\pi(s_t, a_t;\theta_{p,\mathbb{T}}) = \sum_{i=0}^{N-1} \gamma^i r_{t+i}, t \in \{t_0, t_0+N-1\}, \tag{2}$$

where γ is the discount factor, $0 < \gamma < 1$. Then, the advantage of each action taken can be obtained by,

$$A(s_t, a_t;\theta_{p,\mathbb{T}}, \theta_{v,\mathbb{T}}) = Q_\pi(s_t, a_t;\theta_{p,\mathbb{T}}) - V(s_t;\theta_{v,\mathbb{T}}). \tag{3}$$

Equation (3) indicates how much better the actual selected action is than the average.

Note that an episode is defined to consist of N consecutive samples, where N is equal to batch size. This way, all samples needed to calculate the advantage function can be found in the experience buffer [14].

Then, the objective function of policy network $L_{\theta_{p,\mathbb{T}}}$ and the loss function of value network $L_{\theta_{v,\mathbb{T}}}$ can be used to calculate the gradient of the policy and the value network, and then the global parameters $\theta^*_{p,\mathbb{T}}$ and $\theta^*_{v,\mathbb{T}}$ can be updated according to the gradient (line 18). $L_{\theta_{p,\mathbb{T}}}$ and $L_{\theta_{v,\mathbb{T}}}$ can be expressed as follows:

$$\begin{aligned} L_{\theta_{p,\mathbb{T}}} = &- \sum_{t=t_0}^{t_0+N-1} A(s_t, a_t;\theta_{p,\mathbb{T}}, \theta_{v,\mathbb{T}}) \log \pi(a_t|s_t;\theta_{p,\mathbb{T}}) \\ &- \alpha \sum_{t=t_0}^{t_0+N-1} \sum_{a_t \in \{1,2,\cdots,K\}} \pi(a_t|s;\theta_{p,\mathbb{T}}) \log \pi(a_t|s;\theta_{p,\mathbb{T}}), \end{aligned} \tag{4}$$

$$L_{\theta_{v,\mathbb{T}}} = \sum_{t=t_0}^{t_0+N-1} A(s_t, a_t;\theta_{p,\mathbb{T}}, \theta_{v,\mathbb{T}})^2. \tag{5}$$

To increase the diversity of the actions, the second term of $L_{\theta_{p,\mathbb{T}}}$ introduces the policy entropy to improve the agent's ability to explore the environment, and α controls the strength of the entropy regularization term. β and η are the learning rates.

The stopping criterion is that the model has converged. Specifically, we trace the changing of the average blocking probabilities. If the difference between consecutive average blocking probabilities is smaller than a pre-defined threshold, we regard the model to be converged and therefore criterion is satisfied. Through the above steps with Algorithm 1, we train a teacher model that can improve its RMSA policy under a certain task.

Algorithm 1 Training algorithm of the teacher model.

Initialize: experience buffer $D = \phi$, $\varepsilon = 1$.
while not stopping criterion **do**
  Initialize each thread-specific policy network and value network: $\theta_{p,\mathbb{T}} \leftarrow \theta^*_{p,\mathbb{T}}$, $\theta_{v,\mathbb{T}} \leftarrow \theta^*_{v,\mathbb{T}}$.
  while $|D| < 2N - 1$ **do**
    #SAMPLING
    Upon the $TR(v_s, v_d, b)$ arriving, obtain the state s_t.
    Obtain $\pi(a_t|s_t;\theta_{p,\mathbb{T}})$ and $V(s_t;\theta_{v,\mathbb{T}})$ by the policy and the value network.
    With probability ε select an action a_t according to $\pi(a_t|s_t;\theta_{p,\mathbb{T}})$, otherwise $a_t = \text{argmax}_a\{\pi(a_t|s_t,\theta_{p,\mathbb{T}})\}$.
    Obtain reward r_t.
    Store sample $(s_t, a_t, r_t, V(s_t;\theta_{v,\mathbb{T}}))$ in D.
  end while
  #TRAINING
  For the first N samples ($t \in \{t_0, t_0 + N - 1\}$) in the experience buffer D.
  for $t \in \{t_0, t_0 + N - 1\}$ **do**
    Calculate $A(s_t, a_t;\theta_{p,\mathbb{T}}, \theta_{v,\mathbb{T}})$ by Equation (3).
  end for
  Calculate $L_{\theta_{p,\mathbb{T}}}$ and $L_{\theta_{v,\mathbb{T}}}$ by Equations (4) and (5).
  Obtain the policy network and value network gradients $d\theta_{p,\mathbb{T}}$ and $d\theta_{v,\mathbb{T}}$ with $L_{\theta_{p,\mathbb{T}}}$ and $L_{\theta_{v,\mathbb{T}}}$.
  Global parameters $\theta^*_{p,\mathbb{T}}$ and $\theta^*_{v,\mathbb{T}}$ can be updated by: $\theta^*_{p,\mathbb{T}} \leftarrow \theta^*_{p,\mathbb{T}} - \beta d\theta_{v,\mathbb{T}}$ and $\theta^*_{v,\mathbb{T}} \leftarrow \theta^*_{v,\mathbb{T}} - \eta d\theta_{v,\mathbb{T}}$.
  Delete the first N samples in D and set $\varepsilon = \max\{\varepsilon - \varepsilon_0, \varepsilon_{min}\}$.
end while

4.5. Student Model

Due to the similarities between tasks, we try to use the well-trained teacher model to "teach" the student model to learn the optimal RMSA policy for student tasks, as shown in Step 2 of Figure 1. This process is described in more detail in Step 2 of Figure 2. In this way, the student model adjusts its training according to the experience knowledge of the teacher model, in order to expect faster training speed or better performance.

Distillation is a method to transfer experience knowledge from a teacher model $\mathbb{T}$ to a student model $\mathbb{S}$. To transfer the knowledge, a straightforward method is to minimize the distance between the output of the student model and the teacher model. Because the action probability distribution of the output of policy network reflects the learned RMSA policy, we use cross-entropy to fit the output of the two models' policy networks. In order to transfer more knowledge, the teacher model can utilize a relaxed (higher-temperature) softmax than the one used during training [21]. Choose a temperature τ, the outputs of the teacher model's and the student model's policy network are processed by softmax functions to obtain the distributions: $q_\tau(s_t, \theta_{p,\mathbb{T}})$ and $q_\tau(s_t, \theta_{p,\mathbb{S}})$,

$$q_\tau(s_t, \theta_{p,\mathbb{T}}) = \text{softmax}\left(\frac{\pi(a_t|s_t;\theta_{p,\mathbb{T}})}{\tau}\right), \tag{6}$$

$$q_\tau(s_t, \theta_{p,\mathbb{S}}) = \text{softmax}\left(\frac{\pi(a_t|s_t;\theta_{p,\mathbb{S}})}{\tau}\right). \tag{7}$$

The softmax$(\cdot)$ is defined by:

$$\text{softmax}(i) = \frac{e^i}{\sum_j e^j}. \tag{8}$$

Algorithm 2 describes in detail the training process of the student model. The sampling part is same as the teacher model. When the training conditions are met, we first calculate the cumulative discounted reward for each sample (we only consider the first N samples and ignore the discounted reward after N samples) by:

$$Q_\pi(s_t, a_t; \theta_{p,\mathbb{S}}) = \sum_{i=0}^{N-1} \gamma^i r_{t+i}, t \in \{t_0, t_0 + N - 1\} \tag{9}$$

The advantage of each action can be calculated by:

$$A(s_t, a_t; \theta_{p,\mathbb{S}}, \theta_{v,\mathbb{S}}) = Q_\pi(s_t, a_t; \theta_{p,\mathbb{S}}) - V(s_t; \theta_{v,\mathbb{S}}). \tag{10}$$

Let $H(\cdot, \cdot)$ be the cross-entropy function. Then, the similarity between the student model's and the teacher model's policy network can be increased by minimizing the objective function given below:

$$L^{PD}_{\theta_{p,\mathbb{S}}} = \sum_{t=t_0}^{t_0+N-1} H(q_\tau(s_t, \theta_{p,\mathbb{T}}), q_\tau(s_t, \theta_{p,\mathbb{S}})). \tag{11}$$

During the distillation stage, although the value network did not directly obtain the experience knowledge from the teacher model by cross-entropy fitting, the output of the student model's policy network trained via policy distillation affected the generation of the samples, which indirectly affects the training of the value network.

The loss function $L_{\theta_{v,\mathbb{S}}}$ of the student model's value network during distillation is given by:

$$L_{\theta_{v,\mathbb{S}}} = \sum_{t=t_0}^{t_0+N-1} A(s_t, a_t; \theta_{p,\mathbb{S}}, \theta_{v,\mathbb{S}})^2. \tag{12}$$

By optimizing the objective and the loss function above, we can transfer knowledge from the teacher model to the student model.

When the student model is initialized, its DRL agents start from tabula rasa, which means that they have no professional knowledge about the optical network environment of the task, and therefore, they need to learn the optimal RMSA policy by exploring the state and action space for a long time. Therefore, we transfer the knowledge of the teacher model to the poorly performing student model through distillation to reduce ineffective exploration of the student model.

However, although the teacher model is well-trained for the teacher tasks, in the process of policy distillation, its policy has limitations guiding the training of the student model for the student tasks. Therefore, we conduct the policy distillation for the beginning M $TR(s, d, b)$ requests, and then let the student model learn by itself as shown in Step 3 of Figure 2. The objective function and loss function of the first M traffic requests are given by Equations (11) and (12), and the afterward is given by:

$$\begin{aligned} L_{\theta^-_{p,\mathbb{S}}} = &- \sum_{t=t_0}^{t_0+N-1} A(s_t, a_t; \theta^-_{p,\mathbb{S}}, \theta^-_{v,\mathbb{S}}) \log \pi(a_t|s_t; \theta^-_{p,\mathbb{S}}) \\ &- \alpha \sum_{t=t_0}^{t_0+N-1} \sum_{a_t \in \{1,2,\cdots,K\}} \pi(a_t|s; \theta^-_{p,\mathbb{S}}) \log \pi(a_t|s_t; \theta^-_{p,\mathbb{S}}), \end{aligned} \tag{13}$$

$$L_{\theta^-_{v,\mathbb{S}}} = \sum_{t=t_0}^{t_0+N-1} A(s_t, a_t; \theta^-_{p,\mathbb{S}}, \theta^-_{v,\mathbb{S}})^2. \tag{14}$$

where $\theta^-_{p,\mathbb{S}}$ and $\theta^-_{v,\mathbb{S}}$ are the parameters of the policy and the value network of the student model during self-learning, respectively.

Algorithm 2 Training algorithm of student model.

Initialize: experience buffer $D = \phi$, $\varepsilon = 1$.
while not stopping criterion **do**
  Initialize each thread-specific policy network and critic network by: $\theta_{p,\mathbb{S}} \leftarrow \theta^*_{p,\mathbb{S}}$, $\theta_{v,\mathbb{S}} \leftarrow \theta^*_{v,\mathbb{S}}$.
  while $|D| < 2D - 1$ **do**
    #SAMPLING
    Upon the $TR(v_s, v_d, b)$ arriving, obtain the state s_t.
    Obtain $\pi(a_t|s_t;\theta_{p,\mathbb{S}})$ and $V(s_t;\theta_{v,\mathbb{S}})$ by the policy and the value network.
    With probability ε select an action a_t according to $\pi(a_t|s_t;\theta_{p,\mathbb{S}})$, otherwise $a_t = \text{argmax}_a\{\pi(a_t|s_t;\theta_{p,\mathbb{S}})\}$.
    Obtain reward r_t and store sample $(s_t, a_t, r_t, V(s_t;\theta_{v,\mathbb{S}}))$ in D.
  end while
  #TRAINING
  if before M requests **then**
    #DISTILLATION
    For the first N samples ($t \in \{t_0, t_0 + N - 1\}$) in the experience buffer D.
    for $t \in \{t_0, t_0 + N - 1\}$ **do**
      Calculate $A(s_t, a_t; \theta_{p,\mathbb{S}}, \theta_{v,\mathbb{S}})$ by Equation (10).
    end for
    Obtaining training samples $\{(s^j_t, q_\tau(s^j_t, \theta_{p,\mathbb{T}})\}^N_{j=1}$.
    Calculate $L^{PD}_{\theta_{p,\mathbb{S}}}$ by Equation (11) and $L_{\theta_{v,\mathbb{S}}}$ by Equation (12).
    Obtain the policy network and value network gradients $d\theta_{p,\mathbb{S}}$ and $d\theta_{v,\mathbb{S}}$ with $L^{PD}_{\theta_{p,\mathbb{S}}}$, $L_{\theta_{v,\mathbb{S}}}$.
    Global parameters $\theta^*_{p,\mathbb{S}}$ and $\theta^*_{v,\mathbb{S}}$ can be updated by: $\theta^*_{p,\mathbb{S}} \leftarrow \theta^*_{p,\mathbb{S}} - \beta d\theta_{v,\mathbb{S}}$ and $\theta^*_{v,\mathbb{S}} \leftarrow \theta^*_{v,\mathbb{S}} - \eta d\theta_{v,\mathbb{S}}$.
    Delete the first N samples in D and set ε = max{ε - ε_0, ε_{min}}.
  else
    #SELF-LEARNING
    $\theta^{-,*}_{p,\mathbb{S}} = \theta^*_{p,\mathbb{S}}$, $\theta^{-,*}_{v,\mathbb{S}} = \theta^*_{v,\mathbb{S}}$.
    For the first N samples ($t \in \{t_0, t_0 + N - 1\}$) in the experience buffer D.
    for $t \in \{t_0, t_0 + N - 1\}$ **do**
      Calculate $A(s_t, a_t; \theta^-_{p,\mathbb{S}}, \theta^-_{v,\mathbb{S}})$ by Equation (10).
    end for
    Calculate $L_{\theta^-_{p,\mathbb{S}}}$ and $L_{\theta^-_{v,\mathbb{S}}}$ by Equations (13) and (14).
    Obtain the policy network and value network gradients $d\theta^-_{p,\mathbb{S}}$ and $d\theta^-_{v,\mathbb{S}}$ with $L_{\theta^-_{p,\mathbb{S}}}$ and $L_{\theta^-_{v,\mathbb{S}}}$.
    Global parameters $\theta^{-,*}_{p,\mathbb{S}}$ and $\theta^{-,*}_{v,\mathbb{S}}$ can be updated by: $\theta^{-,*}_{p,\mathbb{S}} \leftarrow \theta^{-,*}_{p,\mathbb{S}} - \beta d\theta^-_{v,\mathbb{S}}$ and $\theta^{-,*}_{v,\mathbb{S}} \leftarrow \theta^{-,*}_{v,\mathbb{S}} - \eta d\theta^-_{v,\mathbb{S}}$.
    Delete the first N samples in D and set ε = max{ε - ε_0, ε_{min}}.
  end if
end while

5. Performance Evaluation

In this section, we introduce the simulation results of the proposed policy distillation design with the EONs. We applied the proposed method to three different scenarios: (1) policy distillation between different traffic patterns, (2) policy distillation between different topologies, and (3) policy distillation between different traffic patterns and topologies.

5.1. Parameter Settings

The common parameters used in the simulations are explained in below. For the simulations in Sections 5.2–5.5, these common parameters are used unless otherwise speci-

fied. Moreover, for convenience, the symbols of these key common parameters and their corresponding meanings and values are listed in Table 2.

Table 2. Key parameters and their corresponding meaning and values.

	Notation	Meaning	Value
DRL Environment (i.e., EONs)		Number of frequency slots per link	100
	B	Bandwidth requirement	$[25, 100]$ Gb/s
	K	Number of candidate paths	5
		Bandwidth of a spectrum slot	12.5 GHz
DRL agent	L	Number of hidden layers (teacher model/student model)	8/5
	H	Number of neurons for each hidden layer (teacher model/student model)	256/128
DRL training	γ	Discount rate	0.95
	β/η	Learning rate	1×10^{-5}
	α	Entropy regularization coefficient	0.01
	N	Mini-batch size	200
	M	Number of traffic requests for distillation	100,000
	τ	Temperature	5
	ε_{min}	Final explore rate	0.05

All the topologies used in the simulations are shown in Figure 3, where the weight of each edge of the topology represents the physical length of each link, and they will be used to calculate the FSs in Equation (1). We set the capacity of each fiber link to be 100 FSs. The traffic requests are generated according to independent Poisson processes. In order to ensure that the blocking probabilities of different topologies can fall within a reasonable range, we set a different traffic load for all the different topologies. The traffic patterns and the load for different simulation scenarios will be described in detail later. In addition, the bandwidth requirement of each traffic request is evenly distributed within $[25, 100]$ Gb/s. The number of the shortest paths K is set to be 5, which means the DRL agent is to select a path from 5 candidate paths.

In terms of the neural network architecture, for the teacher model, the policy and value networks both have five hidden layers, with 256 neurons per layer. For the student model, the policy and value networks both have five hidden layers, with 128 neurons per layer. ReLU is used as the activation function for the hidden layers. We set the discount factor γ, the learning rate β and η, the coefficient of the entropy regularization term α, and the temperature of distillation τ to be 0.95, 1×10^{-5}, 1×10^{-5}, 0.01, and 5, respectively. In addition, the number of traffic requests for distillation M is 100,000. During the training, the mini-batch gradient descent algorithm and the Adam optimizer are used, with the mini-batch size N to be 200. The exploration rate ε is set to be 1 at the beginning and gradually decays by ε_0 (set to be 10^{-5}) units during each training process until it reaches ε_{min}, which is 0.05.

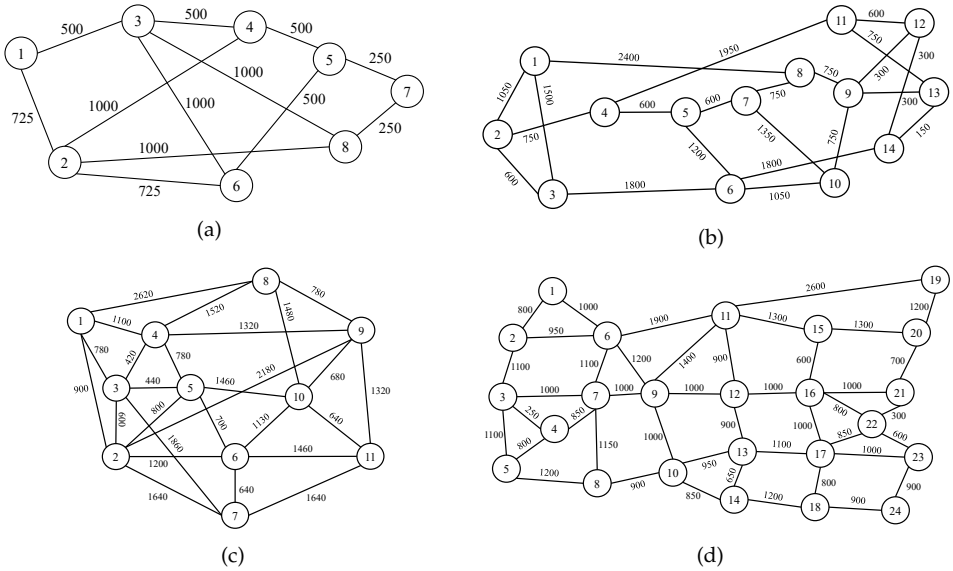


Figure 3. Optical network topologies: (**a**) 8-node, (**b**) 14-node NSFNET, (**c**) 11-node COST 239, and (**d**) 24-node US Backbone.

5.2. Policy Distillation for Different Traffic Patterns

We first evaluate the performance of our proposed scheme for different traffic patterns and the same network topology. In this subsection, both the teacher and the student models are trained over the same network topology: the 14-node NSFNET. The traffic patterns are different. We set the model trained under a uniformly distributed traffic pattern as the teacher model and the model applied for the non-uniformly distributed traffic patterns as the student models.

The traffic pattern is denoted by an $N \times N$ matrix TP, where N(=14) denotes the number of nodes of the NSFNET. The element TP_{ij} represents the traffic load ratio from node i to node j, where $TP_{ij} = 0$ when $i = j$. If TP_{ij} are the same for all i-j pairs ($i \neq j$), the traffic pattern is uniformly distributed. Otherwise, it is non-uniformly distributed. For the student model, we designed three different non-uniform traffic patterns, namely pattern A, pattern B, and pattern C, as shown in Figure 4a,c,e. They correspond to the following three settings:

- Pattern A: non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; $TP_{ij} \neq 0, \forall i \neq j$.
- Pattern B: non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; exist $TP_{ij} = 0$ when $i \neq j$.
- Pattern C: non-uniform; exist $TP_{ij} \neq TP_{ji}$; exist $TP_{ij} = 0$ when $i \neq j$.

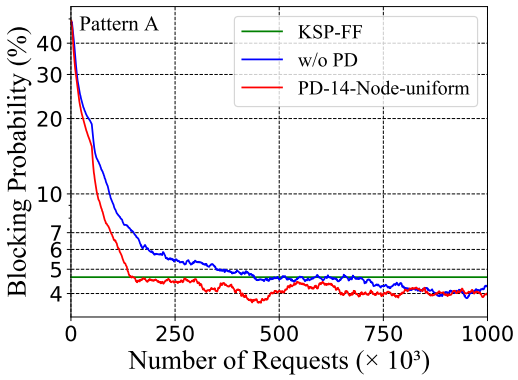
For the uniform traffic patterns, the arrival rate is 12 arrivals per time unit and the average service time is 16 time units, while for the non-uniform traffic pattern, the arrival rate is 16 arrivals per time unit and the average service time is 25 time units. Table 3 records the traffic loads for all the traffic patterns in Section 5.2.

Figure 4b,d,f show the evolution of the simulation results as the number of requests increase, with the blocking probability calculated every 1000 $TR(v_s, v_d, b)$ requests. The blue lines represent the blocking probabilities of the agents learning from scratch without policy distillation ("w/o PD"), while the red lines represent the blocking probabilities of the agents that learn with the policy distilled from the teacher model which is trained with the uniform traffic pattern ("PD-14-Node-uniform"). The green lines represent the blocking probabilities of the baseline algorithm: the K-shortest-path routing and first-fit spectrum allocation (KSP-FF) [36]. The "KSP-FF" in Figure 4b,d,f are the results of applying the KSP-FF algorithm to pattern A, pattern B, and pattern C of the 14-node NSFNET topology, respectively. We can see that, by policy distillation ("PD-14-Node-uniform"), the agent converges faster and achieves lower blocking probabilities, compared to the cases without policy distillation ("w/o PD"). Specifically, the blocking probability reductions are 10%,

10.7%, and 3.6% with pattern A, pattern B, and pattern C, respectively. These results imply that the policy distillation does well in traffic pattern variation tasks.

$$\begin{bmatrix}
0 & 2 & 1 & 1 & 1 & 4 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \\
2 & 0 & 2 & 1 & 8 & 2 & 1 & 5 & 3 & 5 & 1 & 5 & 1 & 4 \\
1 & 2 & 0 & 2 & 3 & 2 & 11 & 20 & 5 & 2 & 1 & 1 & 1 & 2 \\
1 & 1 & 2 & 0 & 1 & 1 & 2 & 1 & 2 & 2 & 1 & 2 & 1 & 2 \\
1 & 8 & 3 & 1 & 0 & 3 & 3 & 7 & 3 & 3 & 1 & 5 & 2 & 5 \\
4 & 2 & 2 & 1 & 3 & 0 & 2 & 1 & 2 & 2 & 1 & 1 & 1 & 2 \\
1 & 1 & 11 & 2 & 3 & 2 & 0 & 9 & 4 & 20 & 1 & 8 & 1 & 4 \\
1 & 5 & 20 & 1 & 7 & 1 & 9 & 0 & 27 & 7 & 2 & 3 & 2 & 4 \\
2 & 3 & 5 & 2 & 3 & 2 & 4 & 27 & 0 & 75 & 2 & 9 & 3 & 1 \\
1 & 5 & 2 & 2 & 3 & 2 & 20 & 7 & 75 & 0 & 1 & 1 & 2 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 0 & 2 & 1 & 61 \\
1 & 5 & 1 & 2 & 5 & 1 & 8 & 3 & 9 & 1 & 2 & 0 & 1 & 81 \\
1 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 3 & 2 & 1 & 1 & 0 & 2 \\
1 & 4 & 2 & 2 & 5 & 2 & 4 & 4 & 1 & 1 & 61 & 81 & 2 & 0
\end{bmatrix}$$

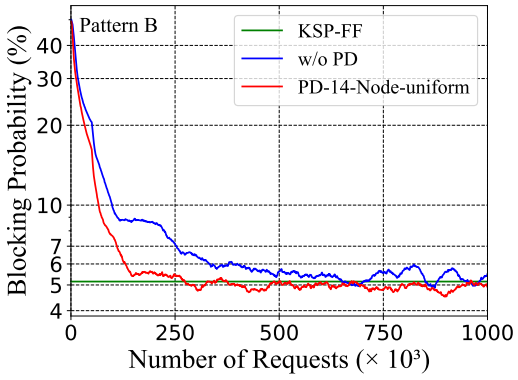
(a) pattern A



(b)

$$\begin{bmatrix}
0 & 2 & 1 & 1 & 1 & 4 & 0 & 1 & 2 & 1 & 0 & 1 & 1 & 1 \\
2 & 0 & 2 & 0 & 8 & 2 & 1 & 5 & 3 & 5 & 1 & 5 & 1 & 0 \\
1 & 2 & 0 & 2 & 3 & 2 & 0 & 20 & 0 & 2 & 0 & 1 & 1 & 2 \\
1 & 0 & 2 & 0 & 1 & 1 & 2 & 1 & 2 & 0 & 1 & 2 & 1 & 2 \\
1 & 8 & 3 & 1 & 0 & 3 & 0 & 7 & 3 & 3 & 1 & 0 & 0 & 5 \\
4 & 2 & 2 & 1 & 3 & 0 & 2 & 1 & 2 & 2 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 0 & 2 & 0 & 9 & 0 & 20 & 0 & 8 & 0 & 4 \\
1 & 5 & 20 & 1 & 7 & 1 & 9 & 0 & 27 & 0 & 2 & 0 & 2 & 4 \\
2 & 3 & 0 & 2 & 3 & 2 & 0 & 27 & 0 & 75 & 2 & 9 & 3 & 0 \\
1 & 5 & 2 & 0 & 3 & 2 & 20 & 0 & 75 & 0 & 1 & 1 & 2 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 2 & 2 & 1 & 0 & 2 & 1 & 61 \\
1 & 5 & 1 & 2 & 0 & 1 & 8 & 0 & 9 & 1 & 2 & 0 & 1 & 8 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 2 & 3 & 2 & 1 & 1 & 0 & 2 \\
1 & 0 & 2 & 2 & 5 & 0 & 4 & 4 & 0 & 1 & 61 & 8 & 2 & 0
\end{bmatrix}$$

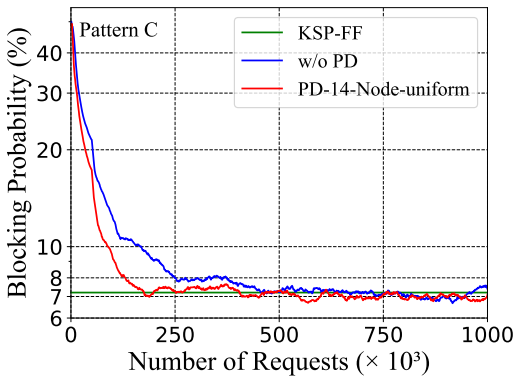
(c) pattern B



(d)

$$\begin{bmatrix}
0 & 9 & 1 & 10 & 1 & 4 & 0 & 1 & 2 & 1 & 0 & 1 & 9 & 1 \\
2 & 0 & 2 & 0 & 8 & 2 & 1 & 5 & 3 & 5 & 1 & 5 & 1 & 0 \\
1 & 2 & 0 & 2 & 3 & 2 & 0 & 20 & 0 & 2 & 12 & 1 & 1 & 2 \\
11 & 0 & 2 & 0 & 1 & 1 & 2 & 1 & 2 & 0 & 1 & 2 & 7 & 2 \\
1 & 8 & 3 & 1 & 0 & 7 & 0 & 7 & 2 & 3 & 1 & 0 & 0 & 5 \\
4 & 2 & 3 & 1 & 3 & 0 & 2 & 1 & 2 & 2 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 2 & 0 & 2 & 0 & 9 & 0 & 18 & 0 & 9 & 0 & 4 \\
6 & 5 & 20 & 1 & 7 & 1 & 9 & 0 & 27 & 0 & 2 & 0 & 2 & 4 \\
2 & 3 & 24 & 2 & 3 & 2 & 0 & 25 & 0 & 75 & 2 & 9 & 3 & 8 \\
1 & 15 & 2 & 0 & 3 & 2 & 20 & 0 & 60 & 0 & 1 & 1 & 2 & 1 \\
0 & 1 & 0 & 11 & 1 & 1 & 0 & 2 & 2 & 1 & 0 & 2 & 1 & 61 \\
1 & 5 & 1 & 2 & 0 & 1 & 8 & 0 & 9 & 1 & 2 & 0 & 1 & 8 \\
1 & 4 & 1 & 1 & 6 & 1 & 6 & 5 & 13 & 21 & 1 & 1 & 0 & 2 \\
1 & 0 & 2 & 2 & 5 & 0 & 4 & 6 & 0 & 1 & 61 & 8 & 2 & 0
\end{bmatrix}$$

(e) pattern C



(f)

Figure 4. (**a**,**c**,**e**): The non-uniform traffic patterns for the student models. (**b**,**d**,**f**): Blocking probabilities under different traffic patterns ((**b**) pattern A, (**d**) pattern B, and (**f**) pattern C) for student model with policy distillation, student model without policy distillation, and the baseline KSP-FF algorithm.

Table 3. Traffic loads for all traffic patterns in Section 5.2.

	Topology	Traffic Pattern	Load
Teacher model	14-node NSFNET	uniform	0.75
Student model	14-node NSFNET	pattern A	0.64
	14-node NSFNET	pattern B	0.64
	14-node NSFNET	pattern C	0.64

5.3. Policy Distillation for Different Topologies

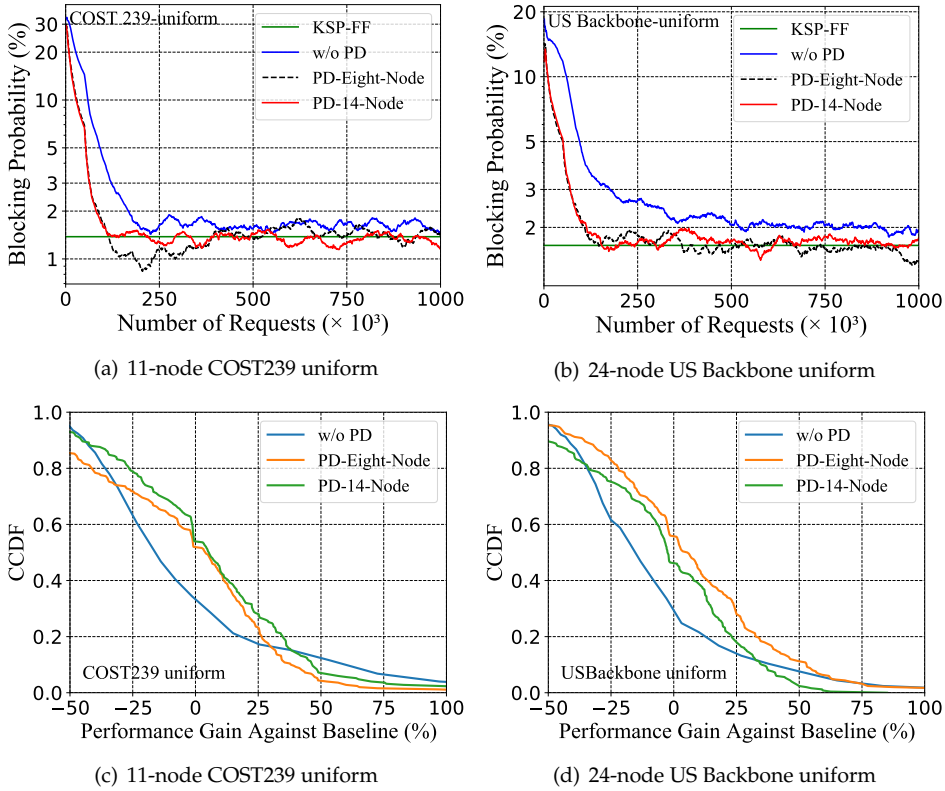
We have also conducted simulations for different topologies to evaluate the performance of the policy distillation scheme. In this case, we train two teacher models in the 8-node topology and the 14-node NSFNET topology, while the other two topologies (the 11-node COST 239 topology and the 24-node US Backbone topology) are used for training the student models. The traffic patterns for all the teacher and student models are the same in terms of distributions: uniform. For the 8-node, 11-node COST239, 14-node NSFNET, and 24-node US Backbone topology, the arrival rate is 14, 16, 12, and 12 arrivals per time unit, and the average service time is 25, 25, 16, and 14 time units, respectively. Table 4 records the traffic loads for all the traffic patterns in Section 5.3.

Table 4. Traffic loads for all traffic patterns in Section 5.3.

	Topology	Traffic Pattern	Load
Teacher model	8-node	uniform	0.56
	14-node NSFNET	uniform	0.75
Student model	11-node COST 239	uniform	0.64
	24-node US Backbone	uniform	0.86

Figure 5a,b show the evolution of the blocking probability by the student models trained in different topologies. We denote the agents that learn with the policy distilled from the teacher models for the 8-node and 14-node NSFNET as "PD-Eight-Node" and "PD-14-Node", respectively. The KSP-FF algorithm is adopted as the baseline, it is applied to the training environment of the uniform distribution 11-node COST239 and 24-node US Backbone topology, respectively, and the results of the "KSP-FF" in Figure 5a,b are obtained. We can observe from Figure 5a that, for the student model trained in the 11-node COST239 topological environment, the cases with policy distillation ("PD-Eight-Node" and "PD-14-Node") reach the performance level of "KSP-FF" faster than the case without the policy distillation ("w/o PD"). Specifically, the blocking performance of the "PD-Eight-Node" and "PD-14-Node" matches that of the "KSP-FF" after about 150,000 and 244,000 traffic requests, but the "w/o PD" consistently performs worse than the "KSP-FF" before 1,000,000 traffic requests.

Similar results are observed in Figure 5b when the student model is trained in the 24-node US Backbone topological environment. Moreover, it can be seen from Figure 5a,b that the cases with the policy distillation ("PD-Eight-Node" and "PD-14-Node") have lower blocking probabilities after convergence compared with the case without the policy distillation ("w/o PD"). These results show that when the topology changes, policy distillation can assist the policy learning in the new environment. Figure 5c,d show the complementary cumulative distribution function (CCDF) with a blocking reduction compared to the "KSF-FF" from different schemes after training with 750,000 traffic requests. For the COST 239 topology, the "PD-Eight-Node" and "PD-14-Node" outperform the "KSP-FF" for around 54% and 52% cases, respectively, while the "w/o PD" only outperforms the "KSP-FF" for around 33% of the cases. For the US Backbone topology, the "PD-Eight-Node" and "PD-14-Node" outperform the "KSP-FF" for around 55.8% and 46.3% of the cases, respectively, while the "w/o PD" outperforms the "KSP-FF" for around 29.5% of the cases. This indicates the effectiveness of policy distillation.



(a) 11-node COST239 uniform

(b) 24-node US Backbone uniform

(c) 11-node COST239 uniform

(d) 24-node US Backbone uniform

Figure 5. (**a**,**b**): Blocking probability in training with different topologies, and (**c**,**d**): complementary cumulative distribution function (CCDF) with blocking reduction compared to KSP-FF algorithm after training with 750,000 traffic requests.

5.4. Policy Distillation for Different Traffic Patterns and Topologies

In this subsection, we change both the traffic patterns and the network topologies for the policy distillation. Similar with Section 5.3, two teacher models are trained under the 8-node topology and the 14-node NSFNET topology, while the student models are applied for the 11-node COST 239 topology and the 24-node US Backbone topology. Besides that, the teacher models are trained under uniform traffic patterns, while the student models are trained under a non-uniform traffic pattern. We have conducted four sets of simulations, denoted as Simulation T-1 to T-4. Detailed simulation settings of the student models are shown in Table 5, and the traffic loads of all the traffic patterns in Section 5.4 are shown in Table 6.

The simulation results are shown in Figure 6a–d. First, we can see that compared with the case without policy distillation ("w/o PD"), taking policy distillation from an eight-node-topology-and-uniform-traffic-pattern teacher ("PD-Eight-Node") and an NSFNET-topology-and-uniform-traffic-pattern teacher ("PD-14-Node") can effectively accelerate the training of student models and obtain lower blocking probabilities for all simulations. Specifically, the "PD-Eight-Node" achieves blocking reductions of 8.3%, 11.9%, 7.8%, and 9.8% for simulations T-1∼T-4, respectively. For the "PD-14-Node", the blocking probability reductions are 7.5%, 11%, 3.9%, and 2.4% for simulations T-1∼T-4, respectively. Meanwhile, Table 7 records the time (approximately) spent by different schemes when the blocking performance reaches the level of the "KSP-FF" in Simulation T-1∼Simulation T-4. In this section, the "KSP-FF" in Figure 6a–d are the results of applying the KSP-FF algorithm to the training environment of Simulation T-1∼Simulation T-4, respectively. We can notice that the "PD-Eight-Node" and "PD-14-Node" learn faster. In Simulation T-1∼Simulation T-4, when the blocking performance reaches that of the KSP-FF, the training time of the "PD-Eight-Node" is reduced by 31.4%, 14%, 57%, and 60.3% compared with that of the "w/o PD", respectively. A similar trend can be seen between the "PD-14-Node" and "w/o PD".

Table 5. Simulation settings for the student models in Section 4.4.

		Student Model
Simulation T-1	Topology	11-node COST239
	Traffic pattern	pattern D (non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; $TP_{ij} \neq 0, \forall i \neq j$)
Simulation T-2	Topology	11-node COST239
	Traffic pattern	pattern E (non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; exist $TP_{ij} = 0$ when $i \neq j$)
Simulation T-3	Topology	24-node US Backbone
	Traffic pattern	pattern F (non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; $TP_{ij} \neq 0, \forall i \neq j$)
Simulation T-4	Topology	24-node US Backbone
	Traffic pattern	pattern G (non-uniform; $TP_{ij} = TP_{ji}, \forall i, j$; exist $TP_{ij} = 0$ when $i \neq j$)

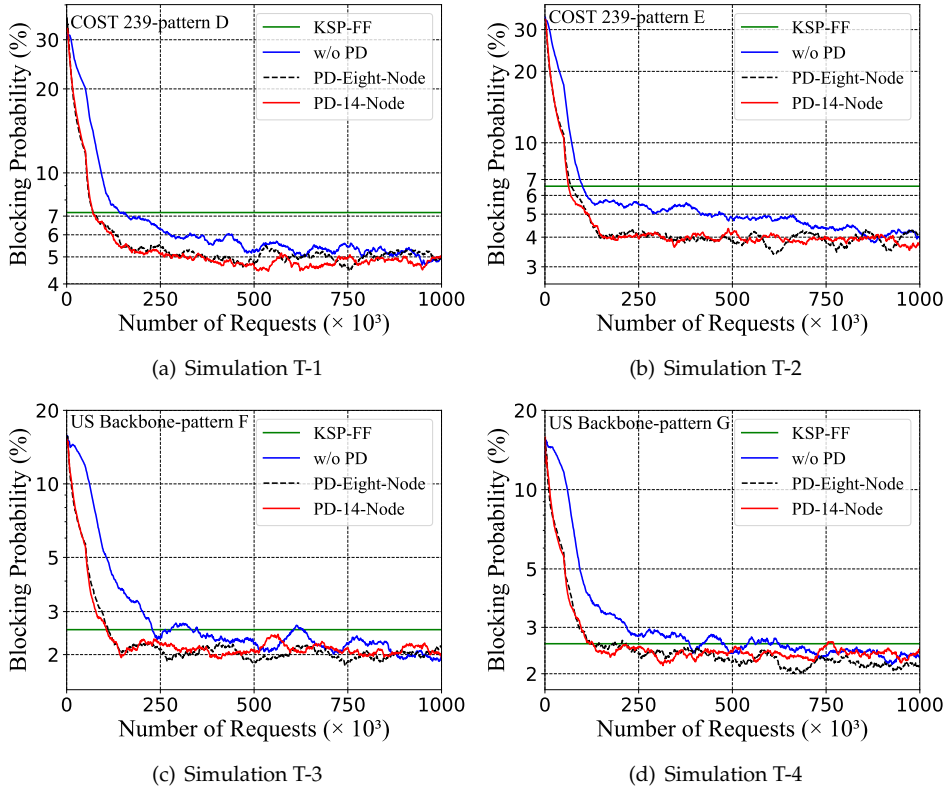
Table 6. Traffic loads for all traffic patterns in Section 5.4.

	Topology	**Traffic Pattern**	**Load**
Teacher model	8-node	uniform	0.56
	14-node NSFNET	uniform	0.75
Student model	11-node COST 239	pattern D	0.64
	11-node COST 239	pattern E	0.64
	24-node US Backbone	pattern F	0.86
	24-node US Backbone	pattern G	0.86

Table 7. Training duration when performance reaches KSP-FF (in seconds).

	"PD-Eight-Node"	**"PD-14-Node"**	**"w/o PD"**
Simulation T-1	1963	1956	2863
Simulation T-2	1939	1895	2258
Simulation T-3	3131	2868	7277
Simulation T-4	3743	3373	9427

For all of the above simulations, we only use the KSP-FF heuristic algorithm as the baseline. As can be seen from the experimental figures, some DRL-based approaches can only achieve a comparable performance with the KSP-FF. For such results, we believe that the performance of the DRL-based approaches is limited by the design of the reward. In this regard, our work [37] has investigated the reward design, and the results are significantly better than the KSP-FF in terms of the blocking probability. However, the focus of this paper is not on the reward design. We pay more attention to the performance comparison before and after the introduction of knowledge distillation. From the above simulations, it can be seen that the blocking performance can be improved by integrating the knowledge distillation method.



(a) Simulation T-1 (b) Simulation T-2

(c) Simulation T-3 (d) Simulation T-4

Figure 6. Blocking probability of different topologies with different non-uniform traffic patterns.

5.5. Policy Distillation with Different Neural Network Size of the Teacher Model

We have also investigated the effect of the size of the teacher model's neural network on the performance of the proposed policy distillation design. Specifically, we design three different neural network settings for the teacher model: (1) three hidden layers with 64 neurons per layer (3 × 64) , (2) five hidden layers with 128 neurons per layer (5 × 128), and (3) eight hidden layers with 258 neurons per layer (8 × 256). The teacher model is trained under the uniform traffic pattern over the 14-node NSFNET, and the student models are trained under the uniform traffic pattern over the COST239 topology. The arrival rate and average service time are the same as in Section 5.2. The results of the blocking probability are shown in Figure 7.

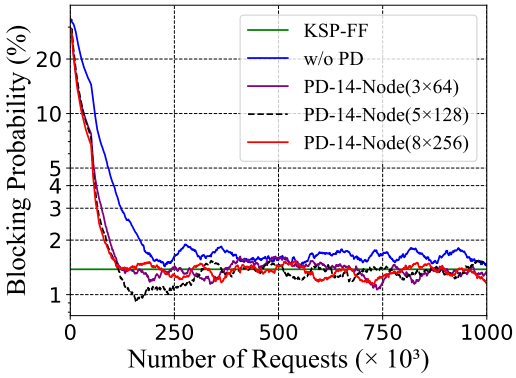


Figure 7. Blocking probability in training with different size of teacher model's neural network.

The result shows that teacher models with different neural network sizes (PD-14-Node (3 × 64), PD-14-Node (5 × 128), and PD-14-Node (8 × 256)) can carry out policy distillation to the student models. This shows that the proposed policy distillation scheme is not limited by the size of the teacher models' neural network. When the neural net-

work architecture of the teacher model and the student model are different, policy learning with policy distillation can also be carried out. This allows knowledge transfer in a broader context.

6. Conclusions

This paper proposes a deep reinforcement learning-based RMSA policy distillation design for the elastic optical networks. It allows the knowledge transfer from a well-trained teacher model under one training environment to a student model under a different environment, so that the training of the latter is accelerated with a better final performance. One highlight is that the student model and the teacher model can be different in terms of the neural network architecture. This allows the knowledge transfer in a broader context. Our method is verified by the simulations of the policy distillation over different traffic patterns and network topologies.

One limitation of our proposal is that the input dimension of the teacher model and the student model must be the same. Recall that the input represents the state of the elastic optical network; the above limitation poses constraints on the state representation. How to break this limitation can be considered for future work. Meanwhile, the performance of the learned RMSA policy in real optical networks should be studied experimentally in future work.

Author Contributions: B.T.: Conceptualization, Methodology, Software, Writing—original draft. Y.-C.H.: Conceptualization, Validation, Writing—review & editing. Y.X.: Conceptualization, Writing—review. W.Z.: Supervision, Writing—review. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62006084), the Basic and Applied Basic Research Foundation of Guangdong Province (2020A1515111110), and the Guangdong Science and Technology Department (2016A010101020, 2016A010101021, and 2016A010101022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their sincere thanks to the Editors and Referees for their enthusiastic guidance and help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. Available online: https://www.cisco.com/c/en_in/index.html (accessed on 1 November 2018).
2. Jinno, M.; Takara, H.; Kozicki, B.; Tsukishima, Y.; Sone, Y.; Matsuoka, S. Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies. *IEEE Commun. Mag.* **2009**, *47*, 66–73. [CrossRef]
3. Gerstel, O.; Jinno, M.; Lord, A.; Yoo, S.B. Elastic optical networking: A new dawn for the optical layer? *IEEE Commun. Mag.* **2012**, *50*, s12–s20. [CrossRef]
4. Zang, H.; Jue, J.P.; Mukherjee, B. A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. *Opt. Netw. Mag.* **2000**, *1*, 47–60.
5. Dinarte, H.A.; Correia, B.V.; Chaves, D.A.; Almeida, R.C. Routing and spectrum assignment: A metaheuristic for hybrid ordering selection in elastic optical networks. *Comput. Netw.* **2021**, *197*, 108287. [CrossRef]
6. Zhang, G.; De Leenheer, M.; Morea, A.; Mukherjee, B. A survey on OFDM-based elastic core optical networking. *IEEE Commun. Surv. Tutor.* **2012**, *15*, 65–87. [CrossRef]
7. Halder, J.; Acharya, T.; Chatterjee, M.; Bhattacharya, U. E-S-RSM-RSA: A novel energy and spectrum efficient regenerator aware multipath based survivable RSA in offline EON. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1451–1466. [CrossRef]
8. Halder, J.; Acharya, T.; Bhattacharya, U. On crosstalk aware energy and spectrum efficient survivable RSCA scheme in offline SDM-EON *J. Netw. Syst. Manag.* **2022**, *30*, 6. [CrossRef]
9. Jia, W.B.; Xu, Z.Q.; Ding, Z.; Wang, K. An efficient routing and spectrum assignment algorithm using prediction for elastic optical networks. In Proceedings of the 2016 International Conference on Information System and Artificial Intelligence (ISAI), Hong Kong, China, 24–26 June 2016; pp. 89–93.

10. Cavalcante, M.; Pereira, H.; Chaves, D.; Almeida, R. Optimizing the cost function of power series routing algorithm for transparent elastic optical networks. *Opt. Switch. Netw.* **2018**, *29*, 57–64. [CrossRef]
11. Harai, H.; Murata, M.; Miyahara, H. Performance of alternate routing methods in all-optical switching networks. In Proceedings of the International Conference on Computer Communications (INFOCOM), Hong Kong, China, 24–26 June 1997; Volume 2, pp. 516–524.
12. Ramamurthy, R.; Mukherjee, B. Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks. *IEEE/ACM Trans. Netw.* **2002**, *10*, 351–367. [CrossRef]
13. Rosa, A.; Cavdar, C.; Carvalho, S.; Costa, J.; Wosinska, L. Spectrum allocation policy modeling for elastic optical networks. In *High Capacity Optical Networks and Emerging/Enabling Technologies (HONET)*; IEEE: Piscataway, NJ, USA; New York, NY, USA, 2012 ; pp. 242–246.
14. Chen, X.; Li, B.; Proietti, R.; Lu, H.; Zhu, Z.; Yoo, S.B. DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks. *J. Light. Technol.* **2019**, *37*, 4155–4163. [CrossRef]
15. Huang, Y.C.; Zhang, J.; Yu, S. Self-learning routing for optical networks. In *Lecture Notes in Computer Science* ; Springer: Berlin/Heidelberg, Germany, 2020. pp. 467–478.
16. Zhao, Z.; Zhao, Y.; Li, Y.; Wang, F.; Li, X.; Han, D.; Zhang, J. Service restoration in multi-modal optical transport networks with reinforcement learning. *Opt. Express* **2021**, *29*, 3825–3840. [CrossRef] [PubMed]
17. Zhao, Y.; Yan, B.; Liu, D.; He, Y.; Wang, D.; Zhang, J. SOON: Self-optimizing optical networks with machine learning. *Opt. Express* **2018**, *26*, 28713–28726. [CrossRef]
18. Xu, L.; Huang, Y.C.; Xue, Y.; Hu, X. Spectrum continuity and contiguity aware state representation for deep reinforcement learning-based routing of EONs. In Proceedings of the IEEE Optoelectronics Global Conference (OGC), Shenzhen, China, 15–18 September 2021; pp. 73–76.
19. Tang, B.; Chen, J.; Huang, Y.C.; Xue, Y.; Zhou, W. Optical network routing by deep reinforcement learning and knowledge distillation. In Proceedings of the Asia Communications and Photonics Conference (ACP), Shanghai, China, 24–27 October 2021; pp. 1–3.
20. Chen, X.; Proietti, R.; Liu, C.Y.; Yoo, S.B. A multi-task-learning-based transfer deep reinforcement learning design for autonomic optical networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2878–2889. [CrossRef]
21. Rusu, A.A.; Colmenarejo, S.G.; Gulcehre, C.; Desjardins, G.; Kirkpatrick, J.; Pascanu, R.; Mnih, V.; Kavukcuoglu, K.; Hadsell, R. Policy distillation. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
22. Gou, J.; Yu, B.; Maybank, S.; Tao, D. Knowledge distillation: a survey. *Int. J. Comput. Vision.* **2021**, *129*, 1789–1819. [CrossRef]
23. Chen, X.; Guo, J.; Zhu, Z.; Proietti, R.; Castro, A.; Yoo, S.B. Deep-RMSA: A deep-reinforcement-learning routing, modulation and spectrum assignment agent for elastic optical networks. In Proceedings of the Optical Fiber Communications Conference and Exposition (OFC), San Diego, CA, USA, 11–15 March 2018; pp. 1–3.
24. Yan, B.; Zhao, Y.; Li, Y.; Yu, X.; Zhang, J.; Wang, Y.; Yan, L.; Rahman, S. Actor-critic-based resource allocation for multi-modal optical networks. In Proceedings of the IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
25. Suárez-Varela, J.; Mestres, A.; Yu, J.; Kuang, L.; Feng, H.; Cabellos-Aparicio, A.; Barlet-Ros, P. Routing in optical transport networks with deep reinforcement learning. *J. Opt. Commun. Netw.* **2019**, *11*, 547–558. [CrossRef]
26. Pujol-Perich, D.; Suárez-Varela, J.; Ferriol, M.; Xiao, S.; Wu, B.; Cabellos-Aparicio, A.; Barlet-Ros, P. IGNNITION: Bridging the gap between graph neural networks and networking systems. *IEEE Netw.* **2021**, *35*, 171–177. [CrossRef]
27. Koch, R.; Kühl, S.; Morais, R.M.; Spinnler, B.; Schairer, W.; Sommernkorn-Krombholz, B.; Pachnicke, S. Reinforcement learning for generalized parameter optimization in elastic optical networks. *J. Light. Technol.* **2022**, *40*, 567–574. [CrossRef]
28. Zhao, Z.; Zhao, Y.; Ma, H.; Li, Y.; Rahman, S.; Han, D.; Zhang, H.; Zhang, J. Cost-efficient routing, modulation, wavelength and port assignment using reinforcement learning in optical transport networks. *Opt. Fiber Technol.* **2021**, *64*, 102571. [CrossRef]
29. Li, B.; Zhu, Z. DeepCoop: Leveraging cooperative DRL agents to achieve scalable network automation for multi-domain SD-EONs. In Proceedings of the Optical Fiber Communication Conference (OFC), San Diego, CA, USA, 8–12 March 2020; p. Th2A-29.
30. Yao, Q.; Yang, H.; Yu, A.; Zhang, J. Transductive transfer learning-based spectrum optimization for resource reservation in seven-core elastic optical networks. *J. Light. Technol.* **2019**, *37*, 4164–4172. [CrossRef]
31. Liu, C.Y.; Chen, X.; Proietti, R.; Yoo, S.B. Evol-TL: Evolutionary transfer learning for QoT estimation in multi-domain networks. In Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 8–12 March 2020; pp. 1–3.
32. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
33. Konda, V.R.; Tsitsiklis, J.N. Actor-critic algorithms. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Denver, CO, USA, 2000; pp. 1008–1014.

34. Kozicki, B.; Takara, H.; Sone, Y.; Watanabe, A.; Jinno, M. Distance-adaptive spectrum allocation in elastic optical path network (SLICE) with bit per symbol adjustment. In Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC), 8–12 March 2020, San Diego, CA, USA, 2010; pp. 1–3.
35. Zhu, Z.; Lu, W.; Zhang, L.; Ansari, N. Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing. *J. Lightw. Technol.* **2012**, *31*, 15–22. [CrossRef]
36. Jinno, M.; Kozicki, B.; Takara, H.; Watanabe, A.; Sone, Y.; Tanaka, T.; Hirano, A. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network. *IEEE Commun. Mag.* **2010**, *48*, 138–145. [CrossRef]
37. Tang, B.; Huang, Y.-C.; Xue, Y.; Zhou, W. Heuristic reward design for deep reinforcement learning-based routing, modulation and spectrum assignment of elastic optical networks. *IEEE Commun. Lett.* **2022**. [CrossRef]