https://doi.org/10.31449/inf.v49i28.8552 Informatica **49** (2025) 137–152 **137**

CSO-MAVNE: A Multi-Agent GAN-GCN Based Collaborative Virtual Network Embedding Model for Dynamic Resource Allocation

Guiyong Sheng*, Li Zhang, Jun Pan, Fujun Wang
Department of Information Engineering, Yangzhou Polytechnic Institute, Yangzhou 225127, China
E-mail: m18952594816_1@163.com, zhangl@ypi.edu.cn, panj@ypi.edu.cn, wangfj@ypi.edu.cn
*Corresponding author

Keywords: virtual network embedding, graph convolutional network, network resource allocation, multi-agent, node

Received: March 10, 2025

To solve the challenges of dynamic topology changes and multi-stage decision-making dependencies in virtual network embedding, a Collaborative Sequential Optimization Multi-Agent Virtual Network Embedding Model (CSO-MAVNE) is proposed. The model integrates three core mechanisms. The topological features of the underlying and virtual networks are extracted through graph convolutional networks. The virtual network topology graph is dynamically constructed and evaluated with the help of generative adversarial networks to optimize the embedding strategy. The distributed collaborative mapping of virtual nodes and links is realized by combining multi-agent reinforcement learning. The Internet Topology Zoo and Synthetic Topology datasets are used for experimental testing with DRL-VNE, MS-GCN, and D-GANE models. The experimental results show that when the number of requests is 300, the embedding success rate of CSO-MAVNE is 82.9%, the cost-benefit ratio is 4.5, the resource utilization rate is 92.5%, and the embedding time is 7.1 s. In the scenario of 10% link failure, the recovery rate remains above 91.7% and the embedding success rate reaches 86.5%. Under high load conditions, the request throughput is 267.9 times/second. The results show that this method is superior to traditional methods in terms of virtual network request embedding success rate, resource utilization rate, and dynamic environment adaptability, and provides a reliable solution for building an efficient and robust resource allocation mechanism.

Povzetek: Razvit je večagentni model CSO-MAVNE, ki združuje GAN, GCN in ojačitveno učenje za učinkovito in robustno razporejanje virov v virtualnih omrežjih z dinamično topologijo.

1 Introduction

With the rapid development of the Internet and the surge of information demand, the traditional network architecture is facing increasingly severe challenges. The continuous emergence of emerging technologies such as the Internet of Things, 5G communication, and cloud computing has led to a rapid increase in network traffic, while also placing higher demands on the flexibility of resource scheduling and management [1]. Traditional network architecture is difficult to adapt to dynamic and changing business requirements and heterogeneous environments due to its fixed hardware resources and rigid configuration [2]. To address these issues, network virtualization technology has gradually become a research hotspot. It abstracts physical resources into multiple logical network slices, providing the possibility of resource sharing and customized services, and providing theoretical and practical support for next-generation network design [3]. In the framework of network virtualization, Virtual Network Embedding (VNE) is the core link of resource allocation, which directly affects the performance and resource utilization rate of the network [4]. The VNE problem involves mapping virtual nodes to physical nodes and allocating virtual links to the paths of physical links, which is essentially a complex combinatorial optimization problem [5]. In recent years, with the development of artificial intelligence technology, significant progress has been made in solving VNE. Zhang W et al. proposed a reproducible multidimensional VNE problem and its solutions to address the scalability and diversity challenges faced by VNEs. Through topology aware preprocessing and embedding methods, in simulation experiments, the average embedding success rate, revenue, and revenue cost ratio were increased by 40.45%, 40.45%, and 299.03%, respectively, and the average cost rate was reduced by 64.16% [6]. Ogino N et al. proposed a distance-based elastic VNE method to address network connectivity issues after critical node failures during the VNE process. Problems were modeled through integer linear programming, and the embedding process was optimized using efficient algorithms. The simulation results showed that VNE solutions with approximate ratios less than 1.2 were derived within 10 seconds [7]. Rezaeimoghaddam P et al. studied the VNE problem in wireless sensor networks and proposed a trust aware algorithm. By adding trust constraints to virtual nodes and links, efficient resource allocation could be achieved while improving network security. The results indicated that this method had significant advantages in throughput, error efficiency, and processing time [8]. Minardi M et al. proposed a dynamic topology aware embedding algorithm and its efficient solution method for the VNE problem in dynamic networks of non-

geostationary orbit satellites. The experiment results showed that the algorithm outperformed the baseline method in terms of computational efficiency, migration cost, and packet loss rate [9].

Graph Convolutional Network (GCN), as a deep learning technique capable of processing graph structured data, has significant advantages in extracting topological information and aggregating node features. Pasa L et al. proposed a new strategy for polynomial graph convolutional layers to address the issue of GCN mainly utilizing local topological information and introducing topological information bias when stacking multiple layers. By independently utilizing neighboring nodes with different topological distances in a single layer, decoupling representations could be generated to avoid the limitations of multi-layer stacking. The results indicated that this method outperformed traditional methods on multiple commonly-used graph classification benchmarks [10]. Bhattacharjee V et al. proposed an enhanced graph representation method to tackle the problem that traditional methods fail to adequately capture the structural similarity of node features in graph data. The method incorporated the concept of overlap into graph data to improve the accuracy of node classification. By inputting the enhanced graph into GCN, this method significantly improved the accuracy in classification tasks [11]. Liu Q et al. proposed a traffic condition prediction method that combined GCN and long short-term memory network. By jointly modeling the urban road topology and traffic flow dynamic characteristics, high-precision prediction of traffic trends was achieved [12]. Li S et al. proposed a dynamic data detection method that combined GCN and temporal convolutional network to address the problem of insufficient accuracy in dynamic data processing under network environment. This method modeled and analyzed network data from two dimensions, time and space, effectively improving the accuracy and processing efficiency of data detection [13].

Combined with Table 1, previous methods mostly use separate processing of node and link embedding, ignoring the coupling between the two in topology and resource dependency, resulting in local optimality and lack of global coordination in the embedding scheme. To overcome the above shortcomings, this study integrates the Generative Adversarial Network (GAN) and GCN for joint modeling, and proposes the Collaborative Generative and Graph Convolutional Network Embedding Algorithm (CGCNE). GCN extracts the structural features of the underlying network and virtual requests, and GAN is used to dynamically generate and optimize the virtual network topology, thereby achieving integrated coordination of node and link mapping and improving the global consistency of embedding and resource allocation efficiency. Finally, the key contents of the above VNE and GCN related methods are detailed in Table 1.

In addition, in the face of dynamic scenarios such as continuous expansion of network scale and frequent node/link failures in actual applications, the centralized embedding strategy performs poorly in terms of timeliness and robustness. To this end, the study further introduces a multi-agent reinforcement learning framework and proposes a Collaborative Sequential Optimization Multi-Agent Virtual Network Embedding Model (CSO-MAVNE). The self-attention mechanism is used to model the temporal dependencies in the embedded tasks, and the joint advantage function is used to coordinate the strategies of each agent to achieve efficient decoupling and collaborative optimization of virtual node and link mapping. This mechanism supports distributed deployment, adapts to dynamic network topology changes, and enhances the system's recovery capability and strategic robustness in fault scenarios.

To address the structural modeling deficiencies and policy coordination performance bottlenecks in VNE, this study proposes a joint modeling method that integrates generative adversarial networks and GCNs, and designs a collaborative decision-making mechanism in combination with multi-agent reinforcement learning to improve the distributed optimization capabilities of the system in resource scheduling tasks. The main innovations include: (1) Introducing GAN and GCN into embedding modeling to improve the collaborative perception capabilities of nodes and links; (2) Constructing a multi-agent architecture based on a joint advantage function to achieve policy distribution optimization and collaborative improvement; (3) Designing a dynamic sequence input mechanism to enhance the model's stability in handling burst requests. Experimental results show that CSO-MAVNE outperforms existing methods in multiple dimensions, including resource utilization, acceptance rate, embedding efficiency, and fault-tolerant recovery.

Table 1: Summary table of related work.

Author	Method	Dataset	Evaluation metrics	Results summary
Zhang W et al. [6]	ReMiDvne	OpenStack platform	Average rates of acceptance ratio,communication costs	Reduce communication costs by up to 45.93 and 63.43 percent
Ogino N [7]	Distance-based resilient VNE	Synthetic topology	Computation time, Approximation ratio	Approximate solution with ratio <1.2 within 10s
Rezaeimoghaddam P et al. [8]	TA-VWSN	WSN	Average network throughput, and processing time	Improved throughput, reduced errors, better efficiency

Minardi M et al. [9]	DTA-VNE	MIRSAT	quantify the packet lost	Reduce the packet lost by ~ 2.5 – 5% compared to baselines.
Pasa L et al. [10]	GNN with single PGC layer	Graph classification benchmarks	Classification accuracy	Outperformed conventional GCNs
Bhattacharjee V et al. [11]	Enhanced graph representation	Real-world graphs	Node classification accuracy	Improved GCN accuracy
Liu Q et al. [12]	LSTM - GCN	Real traffic data	Prediction accuracy, Reliability	High accuracy and adaptability
Li S et al. [13]	Proposed GCN + temporal convolution	Simulated dynamic network data	MAE, MAPE, RMSE, Processing time	Higher accuracy and lower delay

2 Methods and materials

To solve the global optimization problems caused by the decoupling of virtual node and link mapping and improve the system's adaptability in large-scale dynamic environments, the following core model modules are designed. First, to solve the problem of separation of node and link mapping in traditional methods, the CGCNE method (Section 2.1) is proposed to enhance the coordination and structural perception of embedded strategies. Second, to deal with the problems of slow response and poor robustness of centralized strategies, the CSO-MAVNE model (Section 2.2) is constructed to realize distributed decision-making and dynamic adaptation of resource scheduling.

2.1 Joint modeling VNE method based on GAN and GCN

Traditional VNE methods often adopt a staged processing mode, modeling node mapping and link mapping separately, but lack effective decision coordination between the two stages, resulting in insufficient global optimization capability for resource allocation [14-15]. A joint modeling VNE method is proposed to address this issue, which optimizes the mapping decisions of nodes and links simultaneously to avoid segmentation and limitations in the solution space. GAN and GCN are introduced, with their powerful feature extraction capabilities being utilized to capture network topology information and design more coordinated embedding strategies.

To capture the structural dependency and dynamic characteristics between physical links, three interaction relationship modeling methods are constructed: equation (1) describes the resource competition intensity between links, equation (2) describes the link dependency caused by upstream traffic, and equation (3) reflects the degree of coordination between bidirectional links. The three formulas jointly generate a link similarity matrix, which is used as a graph structure prior input to the GCN and GAN modules to guide the network to more accurately perceive the topological structure and its resource status.

Firstly, in virtual network topology, different links may compete with each other due to the traffic allocation of the target node, and their weight matrix is shown in equation (1).

$$A_{E,(u,v),(v,w)} = \exp\left(- \frac{(T^{f+}(u) + T^{f+}(v) - M)^2}{\sigma^2} \right) \quad (1)$$

In equation (1), T is the time window, and $f+$ represents the positive direction link, that is, the direction in which data flows out from the current node. $T^{f+}(u)$ and $T^{f+}(v)$ respectively represent the total outflow of nodes u and v. M is the average traffic output of the node. σ is the standard deviation. Secondly, to capture the mutual influence between upstream and downstream nodes, node correlation is shown in equation (2).

$$A_{E,(u,v),(v,u)} = \exp\left(- \frac{(T^{f-}(u) + T^{f+}(v) - M)^2}{\sigma^2} \right) \quad (2)$$

In equation (2), $f-$ represents the reverse link, that is, the direction in which data flows toward the current node. $T^{f-}(u)$ represents the total incoming traffic of node u to describe the traffic interaction and influence mechanism between nodes. Finally, to characterize the correlation between bidirectional interactions of links, the weight matrix of bidirectional relationships is shown in equation (3).

$$A_{E,(u,v),(v,u)} = \exp\left(- \frac{(T^{f_v^+}(u) + T^{f_u^+}(v) - M)^2}{\sigma^2} \right) \quad (3)$$

In equation (3), $T^{f_v^+}(u)$ represents the traffic from node u to node v, and $T^{f_u^+}(v)$ represents the traffic from node v to node u. By constructing these three link modes, the competition, upstream and downstream, and bidirectional interaction relationships between links can be fully characterized. By combining the feature aggregation capability of GCN, these link features are extracted and fused to optimize the effectiveness of VNE. Subsequently, to further optimize VNE, GAN is introduced to automatically generate and evaluate the effectiveness of the network topology graph, as shown in Figure 1.

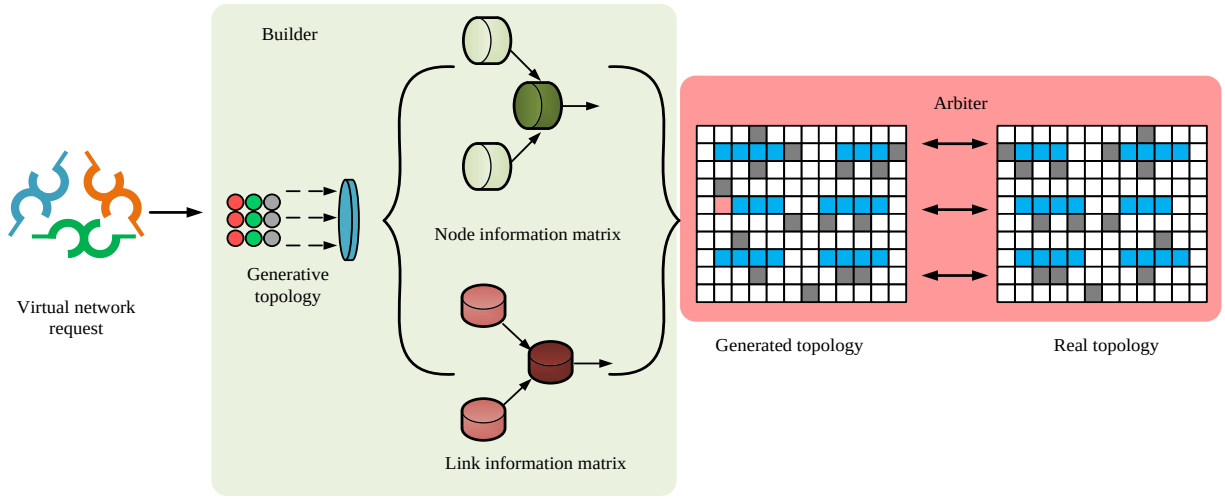


Figure 1: VNE topology generation model based on GAN.

As shown in Figure 1, it includes two core modules: builder G_θ and arbiter D_ϕ. The generator generates node information matrices and link information matrices based on the input network structure characteristics (including node degree distribution, topological connectivity, link bandwidth range, etc.) and virtual request information, which together constitute the simulated topology structure. Next, the generated topology is compared with the actual topology. This comparison result is then fed into the discriminator, which assesses the structural consistency between them. Based on this assessment, the discriminator provides feedback that guides the generator. Through adversarial training, the generator is prompted to continually enhance the structural rationality and distribution diversity of the topologies it produces. The whole process focuses on enhancing the structural authenticity of the generated topology rather than directly evaluating its mapping effectiveness.

The goal of Builder G_θ is to optimize the generated graph structure by minimizing the following objective function, as shown in equation (4) [16].

$$L(\theta) = E_t[\min(r_t(\theta)A_t, clip(r_t(\theta),1-\tau,1+\tau)A_t) + D_\phi(\pi_\theta(a_t|s_t))] \quad (4)$$

In equation (4), $r_t(\theta)$ is an estimate of the dominance function. A_t is the advantage of the state. $\pi_\theta(a_t|s_t)$ is the strategy probability, and $D_\phi(\pi_\theta(a_t|s_t)$ is the weighted loss term output by the discriminator. τ is the temperature coefficient, which is used to control the "smoothness" of the policy distribution and affects the

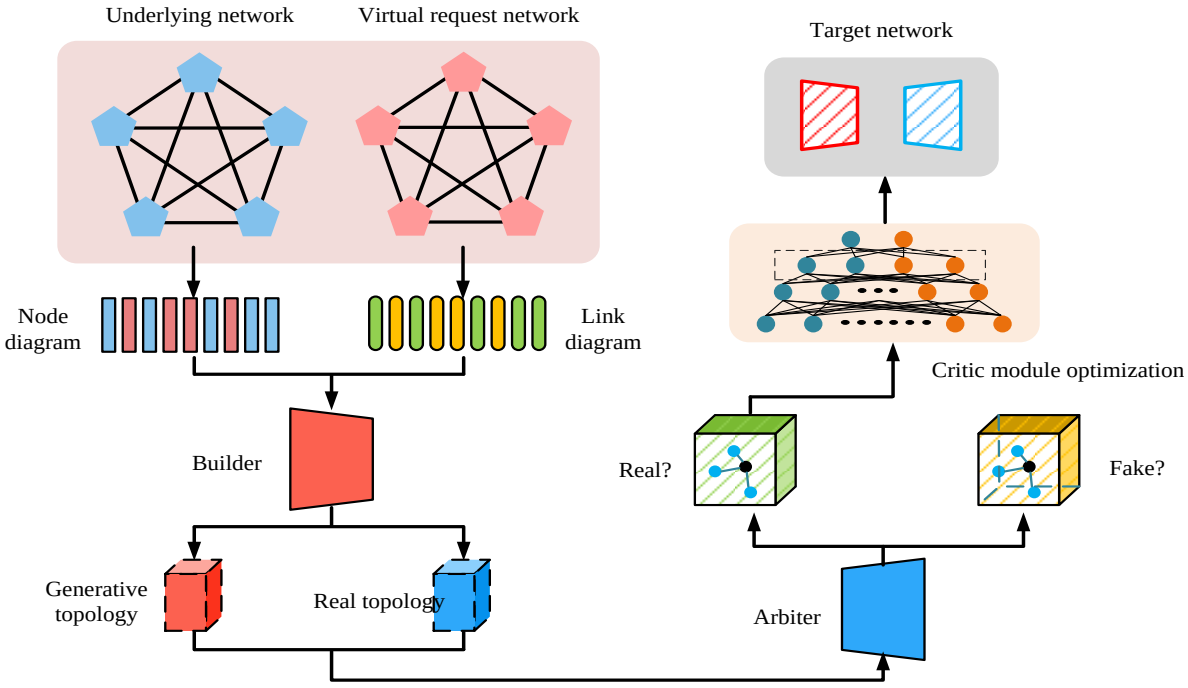


Figure 2: VNE algorithm framework based on CGCNE.

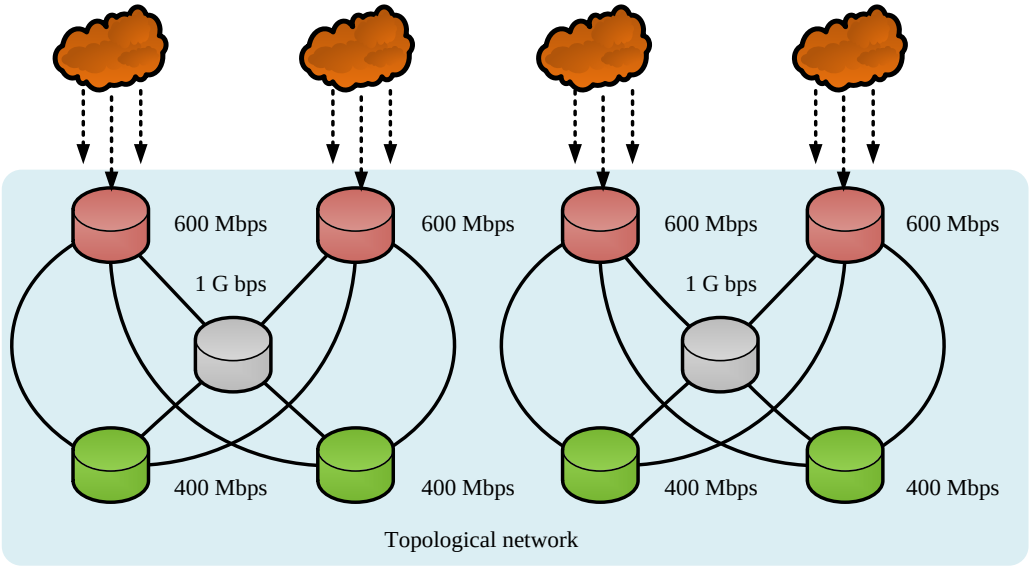


Figure 3: Information flow process of CGCNE framework.

trade-off between exploration and exploitation in policy sampling. A smaller value of τ will make the policy more focused on the dominant action and increase the tendency of deterministic selection; while a larger value of τ encourages the policy to explore more in the action space, thereby improving the generalization ability of the policy. The clipping operation mechanism is introduced in the formula to limit the amplitude of the policy update to avoid training instability caused by excessive updates during the policy optimization process. $clip(r_t(\theta),1-\tau,1+\tau)$ will limit the probability ratio $r_t(\theta)$ within the threshold range to ensure smooth convergence of the optimization process.

Subsequently, arbiter D_ϕ optimizes its ability to judge the true topology by minimizing the objective function, as shown in equation (5) [17-18].

$$L(\phi) = E_{\pi_\theta}\left[D_\phi(\pi_\theta(a_t|s_t))\right] - E_{x\sim P_r}\left[D_\phi(x)\right] \quad (5)$$

In equation (5), P_r is the true data distribution. D_ϕ is used to measure the difference between the generated topology and the real topology. In addition, to improve the learning performance of the builder, the Critic objective is achieved by minimizing empirical error, as shown in equation (6).

$$L_{critic} = E_t\left[(R(s_t,a_t)+\gamma V(s_{t+1})-V(s_t))^2\right] \quad (6)$$

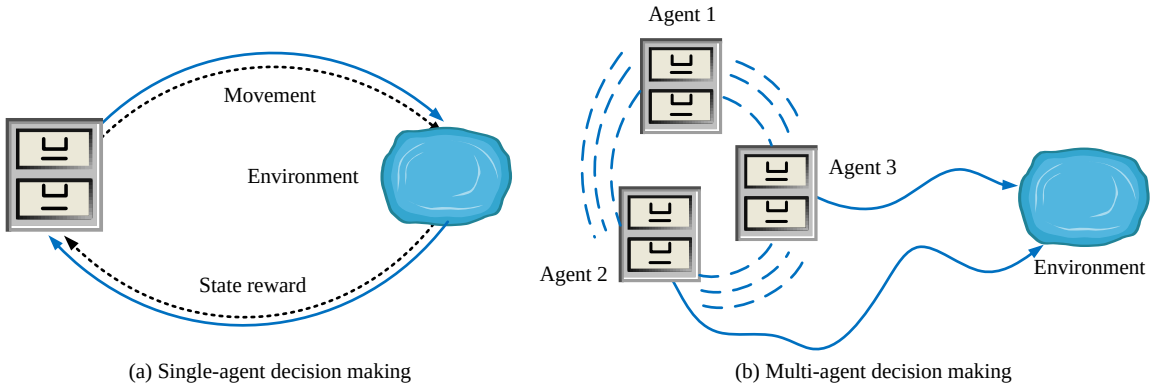
In equation (6), $R(s_t,a_t)$ represents the immediate reward for the current state and action. γ is the discount factor. $V(s_t)$ and $V(s_{t+1})$ are the value functions for the current and next states, respectively. Therefore, based on the above calculations, the VNE algorithm framework based on CGCNE is shown in Figure 2.

As shown in Figure 2, the CGCNE framework first extracts the structural features of virtual requests and the underlying network through GCN to generate a resource requirement matrix for nodes and links. The generator combines these features to generate candidate topologies, the discriminator determines the authenticity of its structure, and the Critic module compares the target topology and optimizes the generation scheme. Finally, the Arbiter combines the feedback from the discriminator and the Critic module to output an embedding strategy that better meets the requirements of the target network. To demonstrate more clearly the specific information transmission and resource allocation process between modules in the CGCNE framework, the information flow diagram is shown in Figure 3.

As shown in Figure 3, the CSO-MAVNE model receives the environment state and generates embedding decisions through a multi-agent structure, where each agent outputs the mapping object of the virtual node or link. After verification and fusion in the Critic module, the mapping relationship is passed to the resource execution module to complete the actual allocation of virtual resources to physical nodes and links. The resource state is refreshed using the mapping and subsequently relayed back to the agent, thus establishing a closed-loop optimization system. This process enables dynamic resource scheduling and adaptive embedding.

2.2 Distributed VNE strategy based on multi-agent reinforcement learning

Although the CGCNE strategy proposed in the previous section can improve the joint embedding quality of nodes and links, its modeling method based on centralized control has problems such as slow response speed and difficulty in policy convergence in large-scale and dynamic environments. In particular, it is difficult to achieve rapid adaptation and concurrent optimization in multi-region resource scheduling [19]. Therefore, based on the previous section, a multi-agent VNE model based on collaborative sequence optimization, CSO-MAVNE, is further proposed. Multiple agents collaborate to embed tasks in a phased manner. First, the node agent completes node mapping based on the physical resource status, and



(a) Single-agent decision making (b) Multi-agent decision making

Figure 4: Schematic diagram of single-agent and multi-agent reinforcement learning.

then the link agent selects the link path based on the node decision. The agents transmit feedback through shared state representation and centralized evaluation modules to achieve collaborative optimization between strategies and improve embedding adaptability. Multi-agent reinforcement learning can effectively adapt to the complexity and real-time requirements of VNE in dynamic environments through the collaboration of multiple agents, as shown in Figure 4. The environment includes resource state input, embedded action output, execution feedback and reward signals. The agent makes embedded decisions based on the current state and continuously optimizes the strategy through the reward mechanism.

In the scenario of a single-agent in Figure 4 (a), the agent selects actions based on the state information of the environment and obtains state rewards through interaction with the environment, thereby optimizing the strategy. In the multi-agent scenario shown in Figure 4 (b), the environmental state is not only influenced by the actions of individual agents, but also by the combined actions of other agents. This interactive characteristic determines that agents need to cooperate in learning to adapt to the globally optimal embedding strategy.

To better capture sequence dependencies in VNE problems, Self Attention Mechanism (SAM) is introduced as the core of multi-agent collaborative optimization in CSO-MAVNE. SAM can effectively handle global dependencies between sequence data, especially for tasks with complex interaction characteristics. As a typical application of self-attention mechanism, the Transformer model maps the input sequence to the latent representation space through the encoder-decoder structure and generates optimization strategies. The calculation is shown in equation (7).

$$Att(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

In equation (7), Q, K, and V represent the query, key, and value, respectively. d_k is the dimension of the key vector, which indicates the feature dimension size of each position in the sequence and determines the information projection space in the attention calculation. The *softmax* function normalizes the dot product result into weights. In multi-agent scenarios, virtual nodes and links are treated as independent sequences, and SAM is used to achieve dependency modeling and global optimization between tasks.

On the basis of SAM, to solve the sequence dependency problem in VNE, the multi-agent strategy further formalizes the embedding task of virtual nodes and links as a sequential decision-making problem. Specifically, each agent is responsible for selecting the optimal strategy for mapping a virtual node or link to the underlying network in the current state. The local priority calculation of the intelligent agent is shown in equation (8).

$$A_{vne}(s,a_n,a_l) = A_{node}(s,a_n) + A_{link(s,a_l)} \quad (8)$$

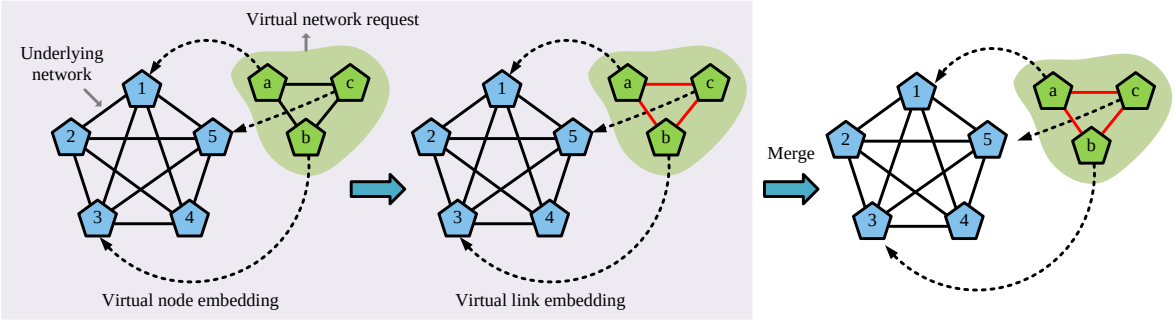


Figure 5: Virtual node and link embedding order decision framework.

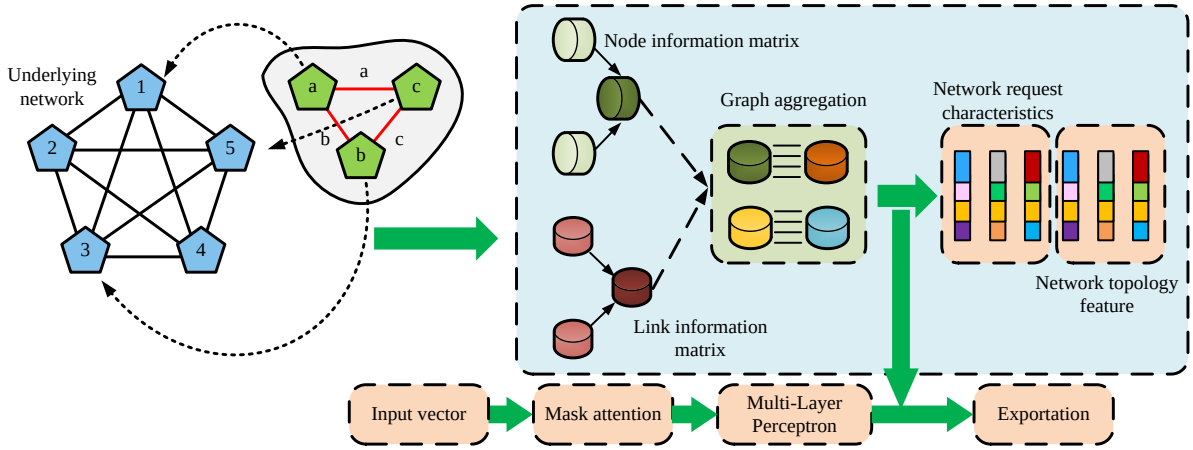


Figure 6: Encoder-decoder architecture for embedding virtual nodes.

In equation (8), s represents the current system operation. a_n and a_l are the embedding actions of virtual nodes and virtual links. $A_{node}(s,a_n)$ and $A_{link(s,a_l)}$ respectively measure the local priority of node and link embeddings. Furthermore, to ensure the collaborative consistency of multiple agents, a joint advantage function is proposed, as shown in equation (9).

$$A_{vne}^{global}(s,a_n,a_l)=\sum_i A_{vne}(s_i,a_n,a_l) \quad (9)$$

In equation (9), s_i represents the current state of agent i. By leveraging the joint advantage function, intelligent agents are able to harmonize their individual decision-making processes while taking into account overarching global objectives. Therefore, the sequential decision-making process and collaborative embedding logic of virtual nodes and links in CSO-MAVNE are shown in Figure 5.

As shown in Figure 5, in the embedding process, the virtual node agent first selects the appropriate physical node for mapping according to the node status. Then, based on the node mapping results, the virtual link agent selects the underlying link resources that meet the bandwidth and path constraints, as shown by the red line in the figure. Finally, the target mapping structure is formed through embedding and merging, providing optimization strategy support for subsequent resource allocation. To achieve efficient mapping of nodes and links in the sequential decision-making framework, an encoder decoder structure is introduced in multi-agent collaboration to dynamically extract network features and generate embedding strategies. The structure is shown in Figure 6.

As shown in Figure 6, for each virtual network request, the encoder extracts the structural features of nodes and links, forms an input vector and sends it to the decoder together with the request features. Initially, the decoder utilizes Mask attention to model the sequential dependencies among nodes. Subsequently, it integrates the structural information generated by the Graph aggregation module, resulting in a fused feature vector. Subsequently, the Multi-Layer Perceptron (MLP) generates the embedding action of the current virtual node based on the vector and outputs the embedding strategy until all nodes are embedded. After all virtual nodes complete the mapping, the decoding process stops and outputs the final embedding scheme. Firstly, the encoder uses the adaptive network feature extraction module GCN to encode the topological features of virtual network requests and underlying physical networks, learning the representational latent space. Encoding parameters are optimized by minimizing experience loss, as shown in equation (10).

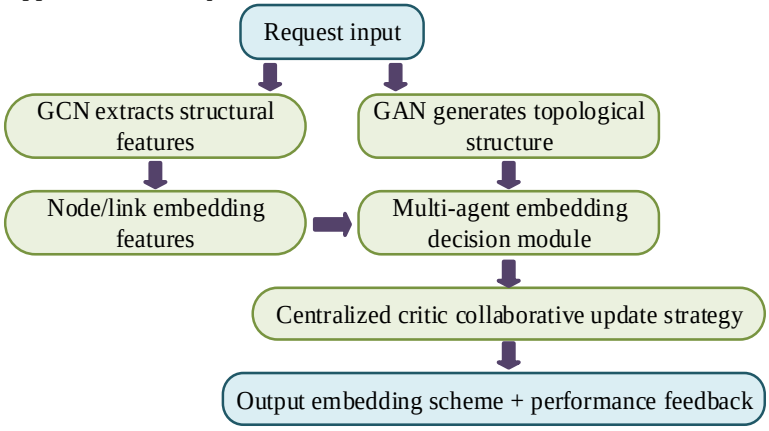


Figure 7: Overall architecture of CSO-MAVNE for VNE.

$$L_{encoder}(\phi)=\frac{1}{T}\sum_{t=0}^{T-1}\left[R(s_t,a_t)+\gamma V_\phi(s_{t+1})-V_\phi(s_t)\right]^2 \quad (10)$$

In equation (10), ϕ is the encoder parameter. $V_\phi(s_t)$ represents the estimated value of the current state. Each time step decoder receives the output features of the encoder and generates a set of embedded action sequences $a^m_{0:M}$ based on the virtual node or link set, maximizing the joint advantage function of the embedding decision, as shown in equation (11).

$$L(\theta)=\frac{1}{2T}\sum_{t=1}^{T-1}\sum_{m=1}^{M}\left[\min(r_t^m(\theta)A_t^m, clip(r_t^m(\theta),1-\tau',1+\tau')A_t^m)\right] \quad (11)$$

In equation (11), A_t^m is the dominant function. τ' controls the strategy generation scope. $r_t^m(\theta)$ is the ratio estimation of the strategy function. Finally, Figure 7 shows its system architecture and main processing stages.

As shown in Figure 7, the overall process of the CSO-MAVNE model includes three stages: feature extraction, candidate topology generation, and multi-agent embedding decision. GCN is used to extract the structural features of nodes and links, and the GAN module generates a well-structured embedding candidate topology. On this basis, a multi-agent reinforcement learning strategy is used to complete the collaborative embedding of nodes and links respectively. This method has the advantages of task decoupling, parallel execution, and strategy sharing, and can effectively improve the resource scheduling efficiency and robustness of the model in large-scale and dynamic environments.

3 Results

3.1 Performance testing of CSO-MAVNE network resource allocation model

The experiment was completed in Windows 11 environment using a desktop computer with 32GB memory and AMD Ryzen 9 5900X processor. The datasets used include typical physical topologies in Internet Topology Zoo and virtual topologies generated by Synthetic Topology based on Waxman model. The former covers the real network structure of operators in multiple regions, and the latter constructs 100 random topologies with a node size of 100–300. The link generation probability was set to 0.3, and the node and link resource requirements were sampled from the uniform distribution U(10,100). The corresponding physical network resources were randomly generated in the range of 1.5–2 times, and all topologies were normalized. Requests were entered into the system in time series order, without batch processing, to simulate the dynamic arrival process. Internet Topology Zoo was used to verify the resource allocation adaptability of the model under the real network structure, and Synthetic Topology was used to evaluate the performance stability and generalization ability of the model under different scales and topological complexities.

For training, the Adam optimizer was uniformly used, and the learning rates of the GCN and the generative adversarial network modules were 0.001 and 0.0002 respectively, and the batch size was 32. GCN used a two-layer graph convolution structure with output dimensions of 64 and 32 respectively, and the activation function was ReLU. The multi-agent strategy adopted the Actor-Critic architecture based on the advantage function, and used the centralized critic to achieve state sharing and independent policy update. Each round of training simulated 300 virtual network requests, with a cumulative number of interaction rounds of 50,000. Performance fluctuations of less than 0.01% for 20 consecutive rounds were considered convergence. The reward function combined resource utilization rate, benefit-cost ratio, and reconstruction overhead, and the basic comparison strategy was to prioritize the maximum residual resources.

To quantify the independent contribution of each key module to the model performance, ablation experiments were carried out, see Table 2.

The results showed that after removing the GAN module, the request throughput dropped to 164.9 req/s and the link utilization increased, indicating that the diversity of the generated structure decreased, resulting in a decrease in embedding efficiency. Removing the joint advantage function increased the number of remappings to 1.44, reflecting its optimization role in multi-agent collaboration. The lack of the self-attention mechanism mainly affected the accuracy of resource scheduling and link pressure distribution.

To verify the generation ability and structural effectiveness of the GAN module in the VNE task, the structural characteristics and training convergence of its output topology were further quantitatively analyzed. The results are shown in Table 3.

Table 2: Ablation study with supplementary metrics.

Model Variant	Request Throughput (req/s)	Avg. Link Utilization (%)	Avg. Remapping Attempts
CSO-MAVNE (Full Model)	183.4	42.1	1.12
w/o GAN (GCN only)	164.9	46.3	1.38
w/o Self-Attention	170.5	44.7	1.26
w/o Joint Advantage Func.	158.7	47.5	1.44

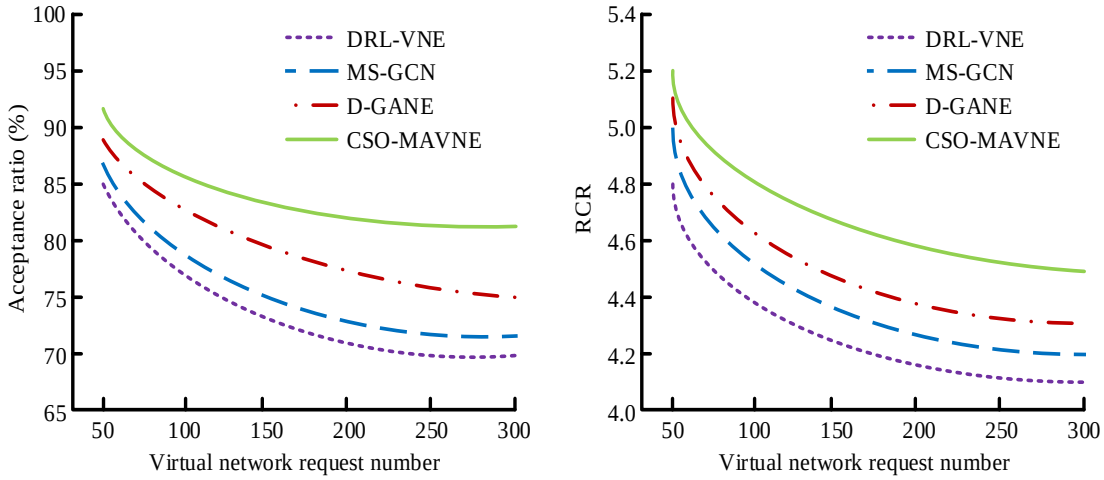
Table 3: Validation results of GAN-Generated topologies.

Metric Type	Real Topologies (Mean)	GAN-Generated Topologies (Mean)	Relative Difference
Average Node Degree	3.85	3.78	-1.80%
Degree Distribution Skewness	0.42	0.39	-7.10%
Average Clustering Coefficient	0.31	0.30	-3.20%
Link Density	0.22	0.23	4.50%

The results showed that the topological structure generated by GAN was highly similar to the real network topology in terms of overall morphology. The differences between key structural indicators such as node average degree, connection density and clustering coefficient and the real topology were all controlled within 5%, indicating that it had high structural fidelity. In addition, the degree distribution skewness was also consistent, indicating that the generated topology could better restore the connectivity pattern of the real network in terms of connection characteristics.

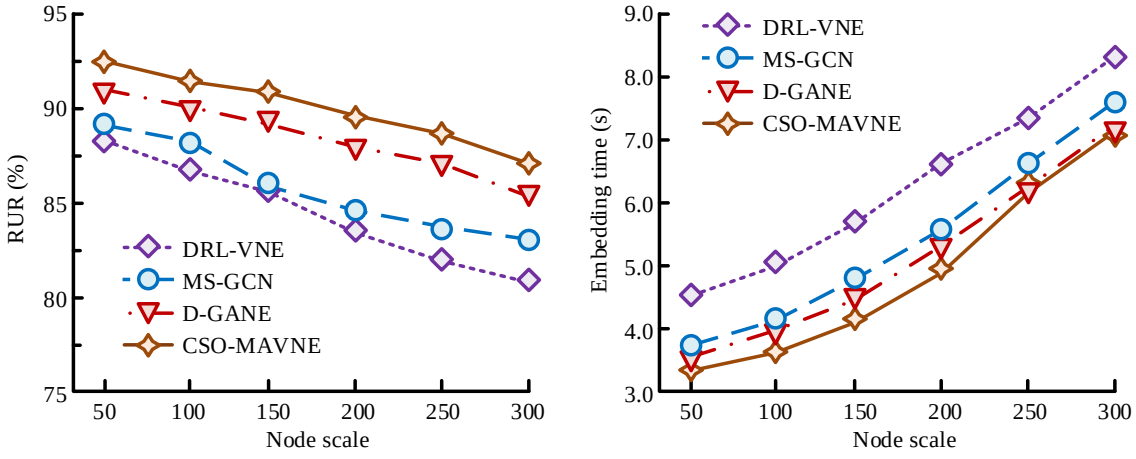
Deep Reinforcement Learning-based VNE (DRL-VNE), Multi-Stage GCN (MS-GCN), and Distributed GAN-based Embedding (D-GANE) were selected as comparative algorithms. DRL-VNE represents the traditional deep reinforcement learning embedding method, MS-GCN integrates graph structure perception capabilities, and D-GANE combines generation mechanism with deep strategy optimization. All three are currently mainstream single-agent VNE algorithms, which are typical and comparable. The bandwidth requirements for the links in the experiment were randomly distributed between 10Mbps and 1Gbps. Firstly, the resource allocation capability and efficiency results under different request scales are shown in Figure 8.

Figures 8(a) and 8(b) show the changes in embedding success rate and Revenue Cost Ratio (RCR) with the number of Virtual Network Requests (VNR), respectively. The number of VNRs refers to the number of virtual network requests received continuously by the system, and the test interval was set to 100 to 300.The embedding success rate is the ratio of successfully completed embedded requests to the total number of requests. The benefit-cost ratio is the ratio between resource benefits and resource overheads (ROs) during the embedding process. The higher the value, the higher the resource utilization rate. In Figure 8(a), when the number of VNRs was 300, the embedding success rates of DRL-VNE, MS-GCN, D-GANE, and CSO-MAVNE were 72.3%, 73.3%, 75.3%, and 82.9%, respectively. In Figure 8(b), when the number of VNRs was 300, the cost-benefit ratios of each model were 4.1, 4.2, 4.4, and 4.5, respectively. CSO-MAVNE utilized the collaborative feature extraction capabilities of GAN and GCN, as well as the collaborative optimization of multiple agents, effectively capturing the dependency relationships between nodes and links. D-GANE performed better in generating features, thanks to the optimization capabilities of its distributed builder. MS-GCN performed slightly worse due to insufficient modeling of topological dependencies. Additionally, it lacks effective global optimization strategies for resource allocation. DRL-VNE had the lowest embedding success rate, and its reinforcement learning model was susceptible to sparse feature interference when facing large-scale requests, making it difficult to adapt to the distributed embedding requirements of complex scenarios. Subsequently, the results of resource utilization rate and embedding time testing are shown in Figure 9.



(a) Embedding success rate test (b) RCR test

Figure 8: Embedding success rate and benefit-cost testing.



(a) RUR test (b) Embedding time test

Figure 9: Resource utilization rate and embedding time testing.

Figures 9(a) and 9(b) show the resource utilization rate and embedding time changes under different numbers of physical network nodes, respectively. Figures 9(a) and 9(b) show the changes in resource utilization rate and embedding time under different node sizes, respectively. Resource utilization refers to the ratio of the allocated node and link resources in the system to the total resources. Embedding time refers to the time required to complete a virtual network request embedding. The shorter the time, the stronger the response capability. In Figure 9(a), when the node size was 300, the resource utilization rates decreased to 81.0%, 83.0%, 85.7%, and 87.2%, respectively. In Figure 9(b), as the node size increased to 300, the embedding time increased to 8.2 seconds, 7.5 seconds, 7.2 seconds, and 7.1 seconds, respectively. The fast strategy generation and collaborative optimization strategy of CSO-MAVNE encoder-decoder structure achieved efficient allocation of global resources. Finally, multiple experimental scenarios were designed, namely static scenario fault free network, dynamic scenario 1 simulating 10% link failure, dynamic scenario 2 simulating 10% link failure, and dynamic scenario 3 simulating 10% node failure. The results are shown in Table 4.

Table 4: Embedding success rate and network stability test results.

Scenario	Index	Model	100 Nodes	200 Nodes	300 Nodes
Scenario 1	Embedding success rate (%)	DRL-VNE	84.3	83.2	82.1
		MS-GCN	87.5	86.5	85.5
		D-GANE	89.8	89.1	88.3
		CSO-MAVNE	92.0	91.2	90.5
Scenario 2	Embedding success rate (%)	DRL-VNE	79.1	78.3	77.8
		MS-GCN	83.2	82.1	81.2
		D-GANE	86.7	85.7	84.5
		CSO-MAVNE	88.5	87.8	86.5
	Recovery rate (%)	DRL-VNE	85.6	84.7	84.3
		MS-GCN	88.9	88.1	87.5
		D-GANE	90.4	89.7	88.5
		CSO-MAVNE	93.2	92.5	91.7
Scenario 3	Embedding success rate (%)	DRL-VNE	76.5	75.4	74.3
		MS-GCN	81.1	80.2	79.4
		D-GANE	84.3	83.5	82.5
		CSO-MAVNE	86.5	85.7	84.3
	Recovery rate (%)	DRL-VNE	89.3	88.5	88.1
		MS-GCN	85.4	84.6	84.3
		D-GANE	87.5	86.8	86.4
		CSO-MAVNE	90.5	89.7	88.5

	RO (%)	DRL-VNE	16.2	16.9	17.5
		MS-GCN	14.3	15.0	15.7
		D-GANE	12.1	12.8	13.5
		CSO-MAVNE	10.5	11.2	11.8

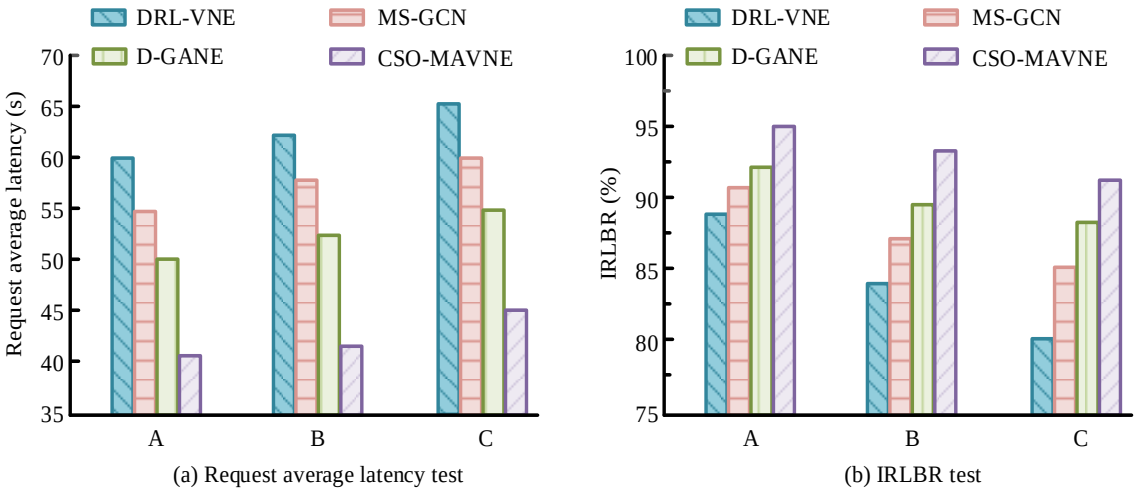
In Table 4, the embedding success rate indicated the proportion of virtual network mappings that were successfully completed under the initial conditions. The recovery rate indicated the proportion of affected requests that were successfully re-embedded after a link or node failure. The RO indicated the proportion of additional resource consumption required to maintain the service during failure recovery.

Under 300 nodes, the embedding success rate of CSO-MAVNE reached 90.5%, and the recovery rates were 91.7% (link failure) and 92.1% (node failure), respectively, with the lowest RO of only 11.8%. The recovery rates of D-GANE were 89.5% and 89.8%, respectively, and the RO was slightly higher. The recovery rate of MS-GCN in dynamic scenarios was maintained at around 85%, and the resource utilization rate was average. DRL-VNE had the lowest recovery rate, only 79.3%, and the highest RO, reaching 17.5%. The dynamic test used link and node failure scenarios to simulate common link interruption and device offline problems, respectively. It had the characteristics of high frequency, strong controllability, and clear interference in real networks, and was more suitable for evaluating the robustness and recovery capabilities of the model in emergency situations.

3.2 Application analysis of CSO-MAVNE network resource allocation model

To verify the optimization ability of the model in cross regional data center resource allocation, the scenario of A, B, and C regional data centers collaborating to process virtual network requests was simulated. Regions A, B, and C each contained 50, 70, and 100 computing nodes, with a total bandwidth of 1 Tbps, 1.5 Tbps, and 2 Tbps, respectively. The experiment randomly generated virtual network requests, which required computing resources, storage resources, and link bandwidth. The model was required to maximize resource allocation efficiency and maintain load balancing between regions while meeting resource requirements. Firstly, the average processing delay and Inter-region Load Balancing Rate (IRLBR) results are shown in Figure 10.

Figures 10(a) and 10(b) show the average processing delay and IRLBR results of each model in three regions, respectively. The average processing delay is the average processing time from the generation of a virtual network request to the completion of mapping. IRLBR was used to measure the balance of resource allocation between different data centers. A higher value indicates a more balanced resource allocation. In Figure 10(a), CSO-MAVNE performed the best in all three regions, with delays of 40ms, 42ms, and 45ms, respectively. The latency of D-GANE and MS-GCN was slightly higher in each region, ranging from 50-55ms and 55-60ms, respectively. DRL-VNE performed the worst, with delays reaching 60ms, 62ms, and 65ms in regions A, B, and C, respectively. In Figure 10(b), the IRLBR of CSO-MAVNE in regions A, B, and C reached 95%, 93%, and 91.5%, respectively. D-GANE performed second best, with IRLBRs of 92.3%, 89.5%, and 88% in the three regions. The equilibrium rate of MS-GCN further decreased, ranging from 90.8% to 85%, while the equilibrium rate of DRL-VNE was the lowest, at 88.5%, 84%, and 80%, respectively. CSO-MAVNE balanced the resource load of each region through a global optimization strategy, while DRL-VNE failed to effectively alleviate resource competition between regions due to policy limitations, resulting in insufficient balance.



(a) Request average latency test (b) IRLBR test

Figure 10: Average processing delay and IRLBR results.

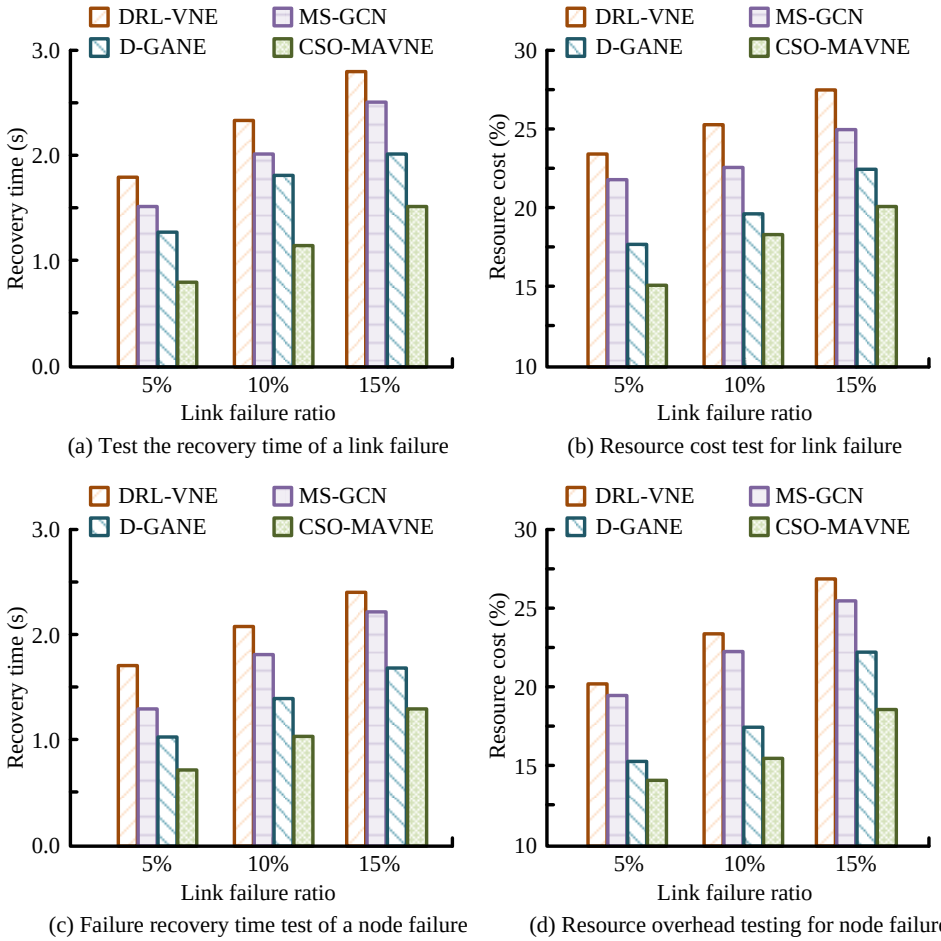


Figure 11: FRT and RO test results under dynamic failure scenarios.

Subsequently, tests were conducted on link failure and node failure in dynamic scenarios to simulate the fault recovery time (FRT) and RO of the model under different failure ratios. Scenario 1 was a dynamic link failure test, with simulated link failure rates of 5%, 10%, and 15%, respectively. Scenario 2 was a dynamic node failure test, with simulated node failure rates of 2%, 5%, and 8%, respectively. Under each condition, the model needed to perform real-time resource remapping, as shown in Figure 11.

Figures 11(a) and 11(b) show the FRT and RO results in link and node failure tests, respectively. FRT was used to measure the time required for the system to restore service from the occurrence of a fault, reflecting the system's response speed to emergencies. RO was used to describe the proportion of additional resources consumed in the process of completing fault repair, reflecting the system resource pressure brought by the fault tolerance mechanism. In Figure 11(a), the FRTs of CSO-MAVNE were 0.8s, 1.1s, and 1.5s, respectively, and the ROs were 15%, 18%, and 20%, respectively. In Figure 11(b), the recovery times of CSO-MAVNE were 0.7s, 1.0s, and 1.3s, respectively, and the ROs were 12%, 15%, and 18%, respectively. The FRT of DRL-VNE was the highest, at 1.6s, 2.0s, and 2.4s respectively, with ROs of 20%, 23%, and 27%, respectively. CSO-MAVNE achieved the lowest FRT and RO through collaborative optimization strategies among multiple agents and efficient dynamic remapping mechanisms. Finally, experiments were conducted to analyze the distribution of request flows and scheduling efficiency under different load conditions. The indicators include Regional Request Distribution Deviation (RDD), Average Latency (AL), Success Allocation Rate (SAR), and Request Throughput Rate (RTR). The results are shown in Table 5.

Table 5: Dynamic request flow distribution and scheduling efficiency results.

Load Level	Model	RDD (%)	AL (s)	SAR (%)	RTR (requests/s)
Low (100/s)	DRL-VNE	22.3	1.82	85.7	85.7
	MS-GCN	18.7	1.53	89.3	89.3
	D-GANE	15.2	1.19	92.8	92.8
	CSO-MAVNE	10.8	1.02	95.6	95.6

Medium (200/s)	DRL-VNE	26.7	2.48	82.1	164.2
	MS-GCN	22.1	2.23	85.4	170.8
	D-GANE	19.1	1.79	88.3	176.6
	CSO-MAVNE	12.6	1.51	91.7	183.4
High (300/s)	DRL-VNE	30.4	3.21	79.4	238.2
	MS-GCN	25.8	2.83	83.2	249.6
	D-GANE	20.7	2.41	86.1	258.3
	CSO-MAVNE	15.3	1.81	89.3	267.9

Table 6: Scalability and time complexity evaluation of different models.

Node Scale	Metric	DRL-VNE	MS-GCN	D-GANE	CSO-MAVNE
500	Throughput	176.2 ± 1.8	180.5 ± 1.5	182.1 ± 1.2	188.7 ± 1.0
	Runtime (s)	5.3	4.8	5	4.3
	P-value (vs CSO)	0.003	0.012	0.025	/
1000	Throughput	168.3 ± 1.5	175.9 ± 1.2	178.0 ± 1.1	183.4 ± 0.9
	Runtime (s)	9.6	8.9	8.3	7.5
	P-value (vs CSO)	0.0012	0.0074	0.019	/
1500	Throughput	151.9 ± 2.0	161.0 ± 1.8	165.7 ± 1.6	172.8 ± 1.2
	Runtime (s)	18.4	17.1	15.8	13.9
	P-value (vs CSO)	0.0008	0.0061	0.011	/

In Table 5, RDD is used to measure whether the request distribution between different regions is balanced. Its value is the standard deviation of the request distribution, expressed in percentage (%). The smaller the value, the more uniform the distribution. AL represents the average time from generation to response of all requests, expressed in seconds (s), reflecting the overall processing efficiency of the system. SAR is used to count the proportion of requests that are successfully mapped under the current load, expressed in percentage (%), which is a reflection of system stability. RTR measures the number of virtual network requests that the system can process per unit time, expressed in requests/s, and is an important indicator for evaluating the system's processing capabilities.

Under low load conditions, the RDD of CSO-MAVNE was 10.8%, AL was 1.02s, SAR reached 95.6%, and RTR was 95.6 request/s. Under medium load, CSO-MAVNE achieved the best overall performance with an RDD of 12.6%, AL of 1.51s, SAR of 91.7%, and RTR of 183.4 requests/s, outperforming all baseline models under the same load conditions. Under high load conditions, the RDD of CSO-MAVNE was 15.3%, AL was 1.81s, SAR was 89.3%, and RTR was 267.9 request/s, all of which were better than D-GANE and MS-GCN, while DRL-VNE performed the worst. CSO-MAVNE benefited from its efficient resource allocation and load balancing strategy, while other models experienced significant performance degradation under high loads.

Finally, to further verify the scalability and efficiency advantages of the proposed model in a larger network, this paper built an embedded test environment at three node scales and introduced a virtual network request flow that simulated the workload of a real data center, referring to the business characteristics of platforms such as Facebook and Alibaba. The experiment conducted multiple rounds of tests on CSO-MAVNE and three comparison models under the same resource configuration and request scale, and the results are shown in Table 6.

As shown in Table 6, CSO-MAVNE showed higher request throughput and lower embedding running time at all node scales, reflecting good scalability and computing efficiency. The results of the independent sample t-test on the throughput difference between CSO-MAVNE and other models showed that at all node scales, CSO-MAVNE reached a significant level of $P< 0.05$ compared with other models, further proving the significant advantages of the proposed model in scalability and processing efficiency.

4 Discussion

Compared with the traffic flow prediction model combining GCN and LSTM proposed by Liu Q et al. in the literature [12], this method mainly focused on time series data modeling under static structure, achieved high prediction accuracy and adaptability through feature fusion, and was suitable for small and medium-sized urban traffic scenarios. When facing the problem of embedding large-scale virtual networks, the CSO-MAVNE model proposed in the study introduced bidirectional link modeling and collaborative multi-agent architecture, which enhanced the model's processing capabilities for link competition, resource coupling, and task parallelism. Performance test results showed that when the number of

VNRs was 300, the acceptance rate of CSO-MAVNE was 82.9%, the benefit-cost ratio was 4.5, the resource utilization rate was 87.2%, and the embedding time was 7.1s.

Compared with the spatiotemporal dynamic data detection method based on GCN and temporal convolutional network proposed by Li S et al. in the literature [13], this study focused on solving the problem of data accuracy degradation in high-frequency dynamic network environments, and achieved a balance between detection accuracy and efficiency through multi-scale modeling. The CSO-MAVNE model in the study focused more on the optimization of resource allocation strategies. Under the guidance of the joint advantage function, the strategy synchronization and global consistency of node and link embedding were achieved. In static scenarios, the embedding success rate of CSO-MAVNE was as high as 90.5%. After the failure of links and nodes, the recovery rate remained above 91.7%. It showed good fault response capability and task continuity guarantee capability.

In summary, the CSO-MAVNE model's performance improvement can be primarily attributed to three key factors: the GCN's ability to embed structural awareness, the GAN-enhanced mechanism for generating feasible topologies, and the optimization of task decoupling execution along with state sharing within the multi-agent collaborative architecture. All the data used in this article were sourced from public simulation networks, did not involve personal privacy or sensitive user information, and posed no ethical or data usage risks.

5 Conclusion

To address the challenges of resource allocation efficiency and complex dynamic topology processing in VNE, a network resource allocation model based on CSO-MAVNE was proposed. In actual application tests, the latency of CSO-MAVNE in three areas was 40ms, 42ms, and 45ms, respectively, and the IRLBR reached 95%, 93%, and 91.5%, respectively. In the dynamic link failure test, the link failure ratios were 5%, 10%, and 15%, respectively, and the FRTs were 0.8s, 1.1s, and 1.5s, respectively, and the ROs were 15%, 18%, and 20%, respectively. In scenarios where the node failure ratios were 2%, 5%, and 8%, respectively, the recovery times were 0.7s, 1.0s, and 1.3s, respectively. Finally, its RDD under high load conditions was only 15.3%, and AL was 1.51s. Experiments results showed that the new method improved the virtual network request acceptance rate, resource utilization, and benefit-cost ratio, providing a reliable basis for optimizing virtual network resource allocation.

Nevertheless, the model may experience communication delays or local strategy convergence problems between multiple agents in ultra-large-scale networks, affecting the overall coordination performance. In addition, due to the extensive use of synthetic topological data during training, there is a potential risk that the model strategy will overfit to a specific structural distribution. In a highly dynamic network environment, the state update of some agents may lag behind the global state change, affecting the real-time performance of embedded decisions. The above problems will be improved in future work by introducing heterogeneous structures, adaptive agent update mechanisms, and expanding real-world data samples.

6 List of nomenclature

CSO-MAVNE: Collaborative Sequential Optimization Multi-Agent Virtual Network Embedding Model
VNE: Virtual Network Embedding
GCN: Graph Convolutional Network
GAN: Generative Adversarial Network
CGCNE: Graph Convolutional Network Embedding Algorithm
SAM: self attention mechanism
MLP: Multi-Layer Perceptron
DRL-VNE: Deep Reinforcement Learning-based VNE
MS-GCN: Multi-Stage GCN
D-GANE: Distributed GAN-based Embedding
RCR: Revenue Cost Ratio
VNR: Virtual Network Requests
IRLBR: Inter-region Load Balancing Rate
FRT: Fault recovery time
RDD: Regional Request Distribution Deviation
AL: Average Latency
SAR: Success Allocation Rate
RTR: Request Throughput Rate
T : Time window
$f+$: Positive directional links
$f-$: Reverse directional links
$T^{f+}(u)$: Total outgoing traffic for node u
M : Average value of outgoing traffic from the node
σ : Standard deviation
$T^{f_v^+}(u)$: Traffic from node u to node v
D_ϕ : Generator
D_ϕ : Discriminator
$r_t(\theta)$: Estimation of the dominance function
A_t : Advantages of Status
$\pi_\theta(a_t|s_t)$: Strategy Probability
$D_\phi(\pi_\theta(a_t|s_t)$: Weighted loss term in the output of the discriminator
τ : Temperature coefficient
P_r : Real data distribution
$R(s_t, a_t)$: Instant rewards for current states and actions
γ : Discount factor
$V(s_t)$: Value function for the current state
Q , K , V : query, key and value
d_k : Dimension of the key vector
s : Current system operations
a_n , a_l : virtual node, virtual link embedding action
$A_{node}(s, a_n)$, $A_{link(s,a_l)}$: Local priority of node, link embedding.
i : The current state of the smartbody i

ϕ : Coder Parameters

$V_\phi(s_t)$: Estimation of the value of the current state

$a_{0:M}^m$: Embedded Action Sequences

A_t^m : Dominance functions

τ' : Control policy update scope

$r_t^m(\theta)$: Ratio estimation of the strategy function

References

[1] Sihan Ma, Haipeng Yao, Tianle Mai, Jingkai Yang, Wenji He, Kaipeng Xue, and Mohsen Guizani. Graph convolutional network aided virtual network embedding for internet of thing. IEEE Transactions on Network Science and Engineering, 10(1):265-274, 2022.https://doi.org/10.1109/TNSE.2022.3207205

[2] Chinmaya Kumar Dehury and Prasan Kumar Sahoo. Failure aware semi-centralized virtual network embedding in cloud computing fat-tree data center networks. IEEE Transactions on Cloud Computing, 10(2):1156-1172, 2022.https://doi.org/10.1109/TCC.2020.2984604

[3] Yang Wang and Qian Hu. A path growing approach to optical virtual network embedding in slice networks. Journal of Lightwave Technology, 39(8):2253-2262, 2021.https://doi.org/10.1109/JLT.2020.3047713

[4] Fujun He and Eiji Oki. Shared protection-based virtual network embedding over elastic optical networks. IEEE Transactions on Network and Service Management, 19(3):2869-2884, 2022.https://doi.org/10.1109/TNSM.2022.3178350

[5] Haipeng Yao, Bo Zhang, Peiying Zhang, Sheng Wu, Chunxiao Jiang, and Song Guo. RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding. IEEE Transactions on Emerging Topics in Computing, 9(2):901-914, 2021.https://doi.org/10.1109/TETC.2018.2871549

[6] Weizhe Zhang, Desheng Wang, Shui Yu, Hui He, and Yan Wang. Repeatable multi-dimensional virtual network embedding in cloud service platform. IEEE Transactions on Services Computing, 15(6):3499-3512, 2021.https://doi.org/10.1109/TSC.2021.3102016

[7] Nagao Ogino. Resilient virtual network embedding ensuring connectivity under substrate node failures. IEICE Transactions on Communications, 105(5):557-568, 2022.https://doi.org/10.1587/transcom.2021EBP3122

[8] Parinaz Rezaeimoghaddam and Irfan Al-Anbagi. Trust-aware virtual network embedding in wireless sensor networks. IEEE Sensors Journal, 23(6):6326-6337, 2023.https://doi.org/10.1109/JSEN.2023.3240386

[9] Mario Minardi, Thang X. Vu, Lei Lei, Christos Politis, and Symeon Chatzinotas. Virtual network embedding for NGSO systems: Algorithmic solution and SDN-testbed validation. IEEE Transactions on Network and Service Management, 20(3):3523-3535, 2022.https://doi.org/10.1109/TNSM.2022.3225748

[10] Luca Pasa, Nicolò Navarin, and Alessandro Sperduti. Polynomial-based graph convolutional neural networks for graph classification. Machine Learning, 111(4):1205-1237, 2022.https://doi.org/10.1007/s10994-021-06098-0

[11] Vandana Bhattacharjee, Raj Sahu, and Amit Dutta. Enhanced graph representations for graph convolutional network models. Multimedia Tools and Applications, 82(7):9649-9666, 2023.https://doi.org/10.1007/s11042-021-11843-7

[12] Quanzhi Liu, Shuang Wu, and Peng Zhang. Statistical analysis of urban traffic flow using deep learning. Informatica, 48(5):23-28, 2024.https://doi.org/10.31449/inf.v48i5.5393

[13] Sheng Li, Mingguang Duan, and Xiaodan Zhou. Dynamic detection method for spatiotemporal data based on hybrid model and singular spectrum analysis. Informatica, 49(12):35-48, 2025.https://doi.org/10.31449/inf.v49i12.7578

[14] Peiying Zhang, Chao Wang, Chunxiao Jiang, Neeraj Kumar, and Qinghua Lu. Resource management and security scheme of ICPSs and IoT based on VNE algorithm. IEEE Internet of Things Journal, 9(22):22071-22080, 2021.https://doi.org/10.1109/JIOT.2021.3068158

[15] Jin Cheng, Yulei Wu, Yeming Lin, Yuepeng E., Fan Tang, and Jingguo Ge. VNE-HRL: A proactive virtual network embedding algorithm based on hierarchical reinforcement learning. IEEE Transactions on Network and Service Management, 18(4):4075-4087, 2021.https://doi.org/10.1109/TNSM.2021.3120297

[16] Huilin Zhou, Huimin Zheng, Qiegen Liu, Jian Liu, and Yuhao Wang. Linear electromagnetic inverse scattering via generative adversarial networks. International Journal of Microwave and Wireless Technologies, 14(9):1168-1176, 2022.https://doi.org/10.1017/S1759078721001331

[17] Peiying Zhang, Chao Wang, Neeraj Kumar, Weishan Zhang, and Lei Liu. Dynamic virtual network embedding algorithm based on graph convolution neural network and reinforcement learning. IEEE Internet of Things Journal, 9(12):9389-9398, 2021.https://doi.org/10.1109/JIOT.2021.3095094

[18] Peiying Zhang, Enqi Wang, Zhihu Luo, Yanxian Bi, Kai Liu, and Jian Wang. Energy-efficient virtual network embedding: A deep reinforcement learning approach based on graph convolutional networks. Electronics, 13(10):1918, 2024.https://doi.org/10.3390/electronics13101918

[19] Stelios Prekas, Panagiotis Karkazis, Vasileios Nikolakakis, and Panagiotis Trakadas. Comprehensive comparison of VNE solutions based on different coordination approaches. InTelecom, 2(4):390-412, 2021.https://doi.org/10.3390/telecom2040023