Following the Compass: LLM-Empowered Intent Translation with Manual Guidance

Lingqi Guo[1,2], Jingyu Wang[1,2*], Jianyu Wu[1,2], Caijun Yan[1,2], Haifeng Sun[1,2*], Zirui Zhuang[1,2], Qi Qi[1,2], Yi Dong[3], Haibao Ren[3], Jianxin Liao[1,2]
[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
[2]E-Byte Technologies Co., Ltd., Beijing, China
[3]Huawei Technologies Co., Ltd., Beijing, China
{guolingqi, wangjingyu, wujianyu, yancaijun, hfsun, zhuangzirui, qiqi8266, liaojx}@bupt.edu.cn,
{dongyi22, renhaibao}@huawei.com

***Abstract*—Intent-Based Networking (IBN) represents a novel paradigm of network automation and intelligence that has gradually been applied to network management. While the emergence of Large Language Models (LLMs) has improved the current state of IBN, hardware heterogeneity and high network dynamics remain significant challenges. Hardware heterogeneity requires that IBN effectively manage a diverse range of devices. The high network dynamics demands that IBN align service needs with rapidly changing network resources. We propose LIT, a framework of LLM-empowered Intent Translation with manual guidance. Given the outstanding language understanding and generation capabilities of LLM, LIT utilizes it in intent translation task. To further address two prevalent problems encountered in IBN, we introduce manual guidance and Mixture of Experts (MoE). Under the guidance of the manual, LLM improves its ability to generate high-quality policies that comply with syntax. After introducing MoE, it makes fine-grained adjustments to the parameters of policies based on network status and service requirements. The experimental outcomes demonstrate that LIT considerably alleviates numerous current challenges confronted by IBN and excels in intent translation, attaining an F1 score that is 56.7% higher than the baseline model.**

***Index Terms*—IBN, LLM, intent translation, MoE**

I. INTRODUCTION

In the grand vision of NextG networks, meeting the demands of highly flexible and diversified business requirements and application scenarios has become an inevitable requirement for the future evolution of wireless networks [1], [2]. The industry is in urgent need of automated tools to assist in service orchestration and device management.

Intent-Based Networking (IBN) is a paradigm that envisions flexible, agile, and simplified network configuration with minimal external intervention [3]–[5]. IBN translates user intents into policies, enabling unified management of heterogeneous devices in dynamic networks. The workflow of intent translation and policy deployed as shown in Fig. 1.

However, due to the limitations of past Natural Language Processing (NLP) technologies, intent translation was often not accurate enough, which directly restricted the widespread application of IBN [6]. RNN and LSTM require a large amount of labeled data for learning, and the transferability is poor [7]. Pretrained Language Models (PLMs), such as BERT [8], lack the capability to understand and generate long texts. This limitation fails to meet the requirements of IBN for fine-grained intent understanding and policy generation.

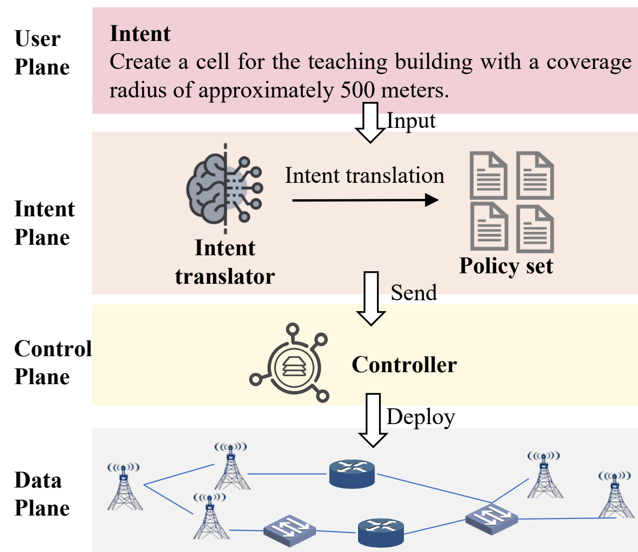


Fig. 1. The workflow of intent translation and policy deployed. The intent is translated into a policy set by the intent translator and then sent to the controller. The controller deploys the policy set to the data plane.

Fortunately, the advanced comprehension and generation capabilities of Large Language Models (LLMs) provide effective solutions to address semantic-level challenges in the realm of intent translation. However, a critical issue remains unaddressed: **how to adapt existing intent translation schemes to the intricate environment of hardware heterogeneity and dynamic networks.** This involves two primary aspects: (1) hardware heterogeneity, where control policies vary across devices from different manufacturers. This diversity necessitates

* Corresponding authors.

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

adapting the intent translation algorithm to accommodate these variations. (2) Network status, where the policies generated must be dynamically adjusted to align with the complex and fluctuating network status. Consequently, within the context of wireless networks, intent translation encounters the following challenges:

- **The lack of intent labeled-data based on heterogeneous hardware.** In most cases, the implementation of intents needs to ensure the grammatical and logical correctness of multiple policies. Currently, there is a lack of labeled data in this field, and hardware manuals have not been effectively utilized.
- **The legality of generated parameter values (PVs).** The intricate relationships between parameters and their corresponding values in policies can give rise to erroneous outcomes, such as a discrepancy between the status of network resources and the specified value range in the manual.
- **The need for generated policies to align with network status.** The generation of metric-level PVs, such as transmission power and bandwidth, must consider user demands and network status. Wireless networks, being more proximate to business users and encountering a wider variety of services, result in diverse requirements.

The intent translation framework, LIT, is introduced to address the aforementioned challenges. It combines manual guidance and the Mixture of Experts (MoE) approach to facilitate the intent translation task. LIT is comprised of two phases: the policy sequence generation (PSG) phase and the parameter value generation (PVG) phase. To ensure the accuracy of intent translation, the context generation capability of LLM is enhanced through manual learning and the Retrieval-Augmented Generation (RAG) approach. Furthermore, we integrate an MoE method based on LLMs, which is adept at understanding users' needs through text-based interactions. This enables more effective coordination of the collaboration of expert models. This approach benefits the LLM in the unified scheduling of experts, solving complex optimization problems, and generating more precise PVs. In comparison to training and optimizing models for each user's requirements individually, this solution significantly reduces the computational resources used for model training and deployment.

In summary, the main contributions of this paper are as follows:

- We propose a novel intent translation framework, LIT, for wireless networks. LIT leverages the understanding and generation capabilities of LLMs to efficiently translate user intents into specific policy sets. LIT integrates manual knowledge and expert experience for fine-grained alignment between business requirements and network states.
- We present an effective generation method that is assisted by manual guidance. This method enhances the LLM's generalization ability for network domain tasks through learning from hardware manuals, which enables the LLM to generate logical and syntactically correct sequences of policies. RAG aids the LLM in understanding the intricate constraints for parameters, which in turn enables the generation of more precise PVs.
- We propose a PV generating method based on MoE, utilizing LLM for precise modeling of user demands, and considering the decisions of multiple expert models to derive the final outcome. The effectiveness of this method is demonstrated by its ability to generate metric-level PVs that impact user service, thereby achieving precise matching between user demands and network resources.

II. RELATED WORK

A. Intent-Based Network

IBN represents a novel network paradigm, characterized by its ability to autonomously convert, verify, deploy, configure, and optimize itself to achieve a desired network state in alignment with user intents [9], [10]. It also possesses the capability to automatically address anomalies, thus ensuring network reliability [11], [12].

Lumi [13] interprets operators' intents through the abstract language Nile [14], translating them into policies. Lumi's templates are fixed and limited, unsuitable for other configuration command languages. [15] proposed CONFPILOT, a retrieval-enhanced policy generation framework that is capable of producing policies with minimal data by consulting manuals. Unfortunately, generating intents should not be confined to the semantic level alone; considering the network resource situation is also crucial. Nassim [16] parses and validates user manuals and configuration files to generate the unified device model for SDN management. This method addresses device heterogeneity but lacks intent-level understanding and integration with network management. Chroma [17] learns the context after the exploitation phase and recommends configuration changes using the contexts for similar locations. Chroma effectively optimizes performance based on past changes but is overly focused on network self-optimization, lacking user interaction. COSYNTH [18] invokes GPT-4 to generate configuration commands with prompts and corrects them using multi-layer validators. COSYNTH efficiently generates and corrects commands but online LLM often lack domain-specific knowledge, failing to ensure successful command generation and lacking consideration of actual network states. [19] developed an intent-based automatic configuration management and orchestration framework that relies on deep learning mechanisms to ensure the scalability and security of resource updates. [20] designed a hierarchical intent processing approach and proposed a mathematical method for fine-grained resource allocation, aligning user expectations with network scenarios. However, the scalability of this solution is limited due to the difficulty of exhaustively capturing corpora related to network services.

In the background of wireless networks, catering to a wide range of commercial user demands introduces significant challenges. These include dealing with more diverse business operations, more extreme resource allocation computations, and more complex device management requirements, posing substantial challenges to intent translation [21], [22]. Our proposed LIT can effectively understand user intent at a semantic level, generate specific configuration templates, and effectively address device heterogeneity through RAG methods. It aims to improve the efficiency of network configuration command execution from multiple QoS metrics perspectives.

B. Large Language Model

Recently, the research on LLM has seen significant advancements, thanks to contributions from both academia and industry. A notable milestone in this journey is the introduction of GPT, which has garnered widespread attention across society [23], [24]. LLMs, known for their exceptional content generation capabilities and transferability, have profoundly impacted various specialized fields [25]. Similar to PLM, LLMs also rely on the Transformer architecture for their model structure, with language modeling as their pretraining objective. However, LLMs are characterized by larger model sizes, training datasets, and overall computational demands compared to PLMs, enabling them to tackle a broader array of complex tasks [26]. Nonetheless, as black-box solutions, LLMs face challenges related to interpretability and security concerns [27], [28].

An approach to address the highlighted concerns is through the application of RAG [29]. The RAG framework augments the capability of models to disseminate precise information by amalgamating external database retrieval processes with generative model operations. The procedural workflow of RAG encompasses two primary phases: Initially, a retrieval mechanism is employed to identify the top-K documents that are most relevant to the given input sequence. Subsequently, these identified documents serve to augment the contextual foundation upon which the LLM generates content. This integrated approach ensures a more accurate and contextually enriched output from the model [30], [31].

C. Mixture of Experts

The MoE framework is a paradigm in machine learning designed to manage and optimize large-scale models by allocating tasks across multiple specialized sub-models. The fundamental appeal of MoE lies in its ability to scale model complexity and handle diverse data types or tasks, which conventional single-model architectures struggle with due to computational and memory constraints.

In this paper, we apply MoE to intent translation task for solving these problems. Prior to this, there has been extensive research based on MoE in the fields of computer vision [32], [33] and NLP [34], [35]. MoE employs a gating mechanism that selects relevant experts from a pool based on the input to make collective decisions. This mechanism allows for the selection of all experts or a sparse combination of them [36]. The advantage of this approach is that the expert pool can be reused across multiple tasks, with only the gating networks needing to be trained specifically for each task [37], [38]. Given the diversity of network services, it is highly impractical to train and deploy separate models for each service or

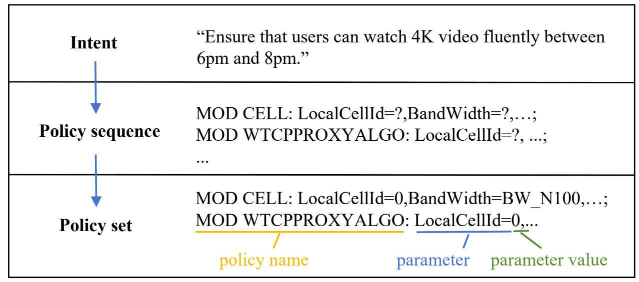


Fig. 2. The example of intent, policy sequence, and policy set. Intent refers to the user's natural language input. Policy set is a set of ordered policies that can be deployed by network controller to achieve the intent. Policy sequence represents a policy set that does not include parameter values.

individual user requirement. Such an approach not only wastes substantial resources used for training and deployment but also results in poor model transferability. Therefore, this paper introduces an MoE method based on LLMs, which selects appropriate expert models for efficient decision-making based on the LLM's understanding of business needs.

III. PROBLEM DESCRIPTION

Generally, IBN translates natural language intents into multiple executable network policies. Intent contains the network operations users anticipate performing on the network, including important information such as time, services, and purpose [39], [40]. An example of two-phase intent translation is shown in Fig. 2. Policy sequence (PS) is an intermediate state between intent and executable policy. It adheres to the policy syntax framework delineated in the manual, though it lacks PVs. The policy set, as the consequence of intent translation, can be deployed. Each policy is composed of a policy name, several parameters, and their values. For each policy, the manual typically includes the policy name, a description of the policy, the parameters included in the policy, descriptions of the parameters, and the range of PVs. The majority of PVs are of types string, int, or checkbox. Additionally, there are two special cases: (1) PVs are constrained by other parameters; (2) PVs are influenced by the network status. Consequently, stringent restrictions on PVs emerge as one of the key factors limiting the quality of intent translation.

Parameters can be classified into two types based on their relation to Quality of Service (QoS): one is semantic-level parameters, such as $CELL\_ID$, $CELL\_NAME$; the other is metric-level parameters, such as $BANDWIDTH$, $POWER$.

In order to achieve greater precision in the generation of metric-level PVs, it is necessary to consider the downstream network status for calculation purposes, such as bandwidth. The notation of the variables can be found in TABLE I. The downstream network topology is defined as: $G = \{D, L\}$, where D represents the set of $|D|$ nodes, L represents the set of $|L|$ links. Each link l ($l \in L$) has a bandwidth resource B_l. The network endeavours to fulfil all user services U by allocating

TABLE I
NOTIONS OF VARIABLES

Symbol	Description
$G(D, L)$	A downstream network topology consists of D nodes and L links.
U	All services.
L_{n_u,d_u}	The set of all links along the path between the source node n_u and destination node d_u.
B_l	The bandwidth resource of link l.
B_u	Bandwidth resources allocated to u.
B_k	k available bandwidth value settings.
T	Various QoS indicators contained in u.
t_n	The individual QoS indicator.
Z_u	Whether the currently allocated bandwidth meets all. QoS indicator requirements of u.
R_{u,t_n}	The benefit of t_n in each u.
$M(\cdot)$	Utility function of standardized evaluation framework.
$\mathcal{T}_n(\cdot)$	The calculation method of t_n.
S_u	Resource condition of links passed by the user to the server.
A	The action space.
W	The weight matrix to the selected n expert models.
$Q(\cdot)$	Q value vector based on the input network state.
$Q^*(\cdot)$	The final weighted Q value vector.

link bandwidth resources B_u to the user service u ($u \in U$). Each user service u includes the source node n_u and the destination node d_u. L_{n_u,d_u} denotes the set of all links along the path between the two nodes. The various QoS indicators contained in the service u are represented by T = $\{t_1$,t_2,...,$t_n\}$. The network generates rewards based on the overall of the Service Level Agreement (SLA) Satisfaction Ratio (SSR) and QoS benefits accrued after providing services to multiple users. In order to calculate SSR, a binary variable $Z_u \in \{0, 1\}$ is defined to indicate whether the currently allocated bandwidth meets all QoS indicator requirements of the service. Therefore, the overall SSR of the network is defined as follows:

$$SSR = \frac{\sum\limits_{u \in U} Z_u}{|U|} \tag{1}$$

Concurrently, the allocation of bandwidth resources will result in a certain degree of link loss. This loss will gradually increase as the link bandwidth resources become constrained. Based on the aforementioned considerations, we define the method in which the network calculates the benefit of each individual QoS indicator t_n in each service requirement u:

$$R_{u,t_n} = \beta_1 M(\mathcal{T}_n(B_u, S_u)) - \beta_2 \sum_{l \in L_{n_u,d_u}} \frac{B_u}{B_l} \tag{2}$$

where β_1 represents the benefits that can be obtained by meeting QoS indicators. $M(\cdot)$ is a utility function of a standardized evaluation framework that maps various QoS indicators to user satisfaction and corresponding benefits. $\mathcal{T}_n(\cdot)$ is the calculation method of QoS indicator t_n. S_u is the resource condition of the links passed by the user to the server. β_2 is the cost coefficient, reflecting the link loss of the network when allocating bandwidth. The upper limit of the QoS indicator t_n that the current network can provide to the user is defined as $t_{n,max}$. Conversely, the lowest QoS that the service can accept is defined as $t_{n,min}$. Therefore, the utility function $M(\cdot)$ can be calculated as follows:

$$M(\mathcal{T}_n(B_u, S_u)) = \frac{\mathcal{T}_n(B_u, S_u) - t_{n,min}}{t_{n,max} - t_{n,min}} \tag{3}$$

In summary, the overall network benefits are defined by the following optimization function:

$$\max_{\boldsymbol{B_u}} \quad \alpha_1 SSR \cdot r + \alpha_2 \sum_{u \in U} \sum_{t_n \in T} R_{u,t_n} \tag{4a}$$

$$\text{s.t.} \quad B_u \leq B_l, \forall l \in L_{n_u,d_u} \tag{4b}$$

$$B_u \in B_k \tag{4c}$$

$$\mathcal{T}_n(B_u, S_u) \geq t_{n,min} \tag{4d}$$

where α_1 and α_2 are coefficients that reflect the relative importance of SSR rewards and QoS benefits. r is the reward obtained based on SSR. B_k=$\{b_1$,b_2,...,$b_k\}$ represents the available bandwidth value settings. The objective of Eq. (4b) is to ensure that the allocated bandwidth does not exceed the residual resources available on each link. Eq. (4c) regulates the range of optional bandwidth parameters. Eq. (4d) ensures that the bandwidth selected by each expert in MoE needs to meet the specific QoS requirement in the service. In order to ensure that network resources do not conflict and to maximize network benefits, it is necessary to allocate an appropriate bandwidth.

IV. METHOD

A. Overview of System

The main architecture of LIT is shown in Fig. 3, which mainly includes two phases: PSG phase and PVG phase. In the PSG phase, we use LLM trained on manuals to translate intent into PS. In the PVG phase, we generate values for two types of parameters by introducing RAG and MoE. Ultimately, this process yields an executable policy set for deployment.

LIT effectively addresses challenges associated with alterations in hardware environments by leveraging manual to generate policies. By thoroughly mining user intent and accurately computing downstream network resources, LIT harmonizes business objectives with network status. This facilitates fine-grained intent translation, thereby enabling the provision of on-demand services.

B. PSG Phase

In IBN, the implementation of an intent often necessitates the simultaneous enactment of multiple policies, which must be logically coherent [41]. To illustrate, if the intent is ”Create a new cell for student dormitories, mainly for watching ultra-clear videos and playing games,” the generated policies would include the creation of a cell, the modification of cell configuration, the activation of the cell, and so forth. Moreover, the generation sequence of these policies must be correct, as incorrect sequencing could result in significant errors when policies are deployed.

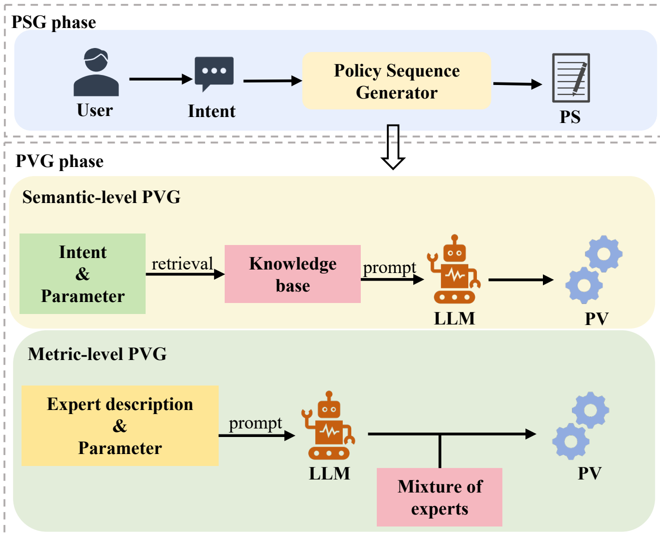


Fig. 3. The main architecture of LIT. After the user propose an intent, the Policy Sequence Generator outputs the PS. Depending on the types of the parameters, different Parameter Value Generators generate the PVs.

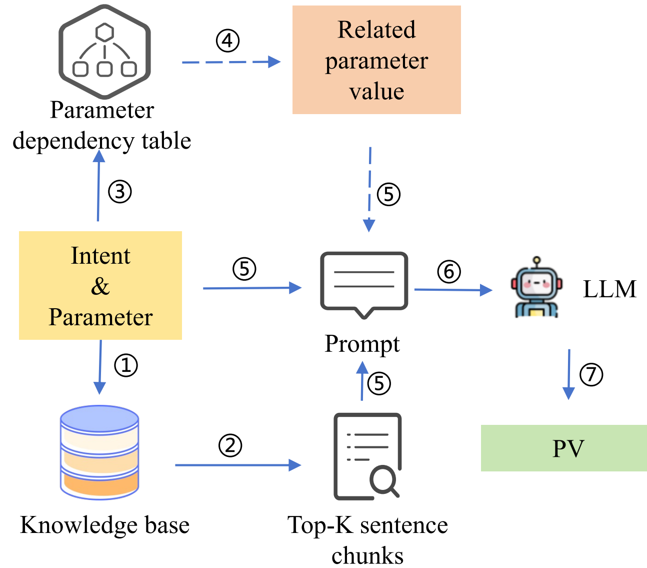


Fig. 4. The workflow of semantic-level PVG.

Consequently, we introduce an offline learning-based Policy Generator approach. Considering the excellent context understanding ability of LLM, we select it as the main component of Policy Sequence Generator. We have comprehensively parsed the manuals, organizing new corpus data to fine-tune LLM. To align with users' usage habits, we have organized the manual data into Q&A format. The dataset obtained from parsing the manual includes knowledge such as descriptions of policies and parameters, and the relationships between policies and parameters. Generating PS based on intent is the target task of the Policy Sequence Generator, while the other datasets serve as auxiliary tasks to help improve the model's performance on PSG.

Following offline fine-tuning with the manuals, the LLM is deployed online to generate PS based on intent. The LLM benefits from the learning of the manuals, enabling it to effectively transform the captured contextual information of intent into logical PS [42].

C. PVG Phase

In intent translation, generating PVs presents a challenging problem. Previous research has defined PVG as a semantic-level task, yet this characterization is not entirely accurate. Given that PVs are not solely related to intents or the syntax rules of manuals, some parameters are involved in QoS, which can be modeled as resource allocation tasks [43], [44]. The parameters in the policy can be divided into two types based on their characteristics. One type is relatively static semantic-level parameters that are independent of QoS. The other type is service-related in intents, and its value depends on not only service requirements but also network status. PVG primarily consists of two components: a semantic-level PVG algorithm based on RAG and a metric-level PVG algorithm based on MoE.

1) Semantic-level PVG based on RAG: The values of semantic-level parameters are typically generated end-to-end by language models. However, the manual imposes strict restrictions on the value of parameters, such as the range of values and the type of value. Consequently, to generate such PVs with greater accuracy, it is necessary to introduce manual knowledge. RAG is introduced as an effective solution to enhance the generation capabilities of LLMs. It enhances the accuracy of PVG by querying related parameter knowledge, such as range and type, from the manual to supplement the prompt.

In addition, the semantic-level parameters often have intricate dependencies, for example, when parameter A takes the value of 1000–2000, parameter B has a range of 500-1000. To solve this problem, the parameter dependency table is manually established to use rules to query whether the target parameter has a dependency relationship with other parameters.

The workflow of semantic-level PVG is shown in Fig. 4. In steps ① and ②, the intent and target parameter serve as query to retrieve the top-k text chunks related to the parameter from the Knowledge Base. In step ③, the Parameter dependency table is used to check whether the target parameter is dependent on any other parameters. If so, the values of the related parameters are output in step ④. In step ⑤, the prompt is formed by the parameter, the related sentence chunks, and the values of the related parameters. In steps ⑥ and ⑦, the formed prompt is input into the LLM, which output the PV.

The text stored in the knowledge base primarily consists of hardware manuals, and it chunked by GPT-4. Inspired by [45], a hierarchical knowledge base can significantly enhance the accuracy and efficiency of retrieval. Given that manuals

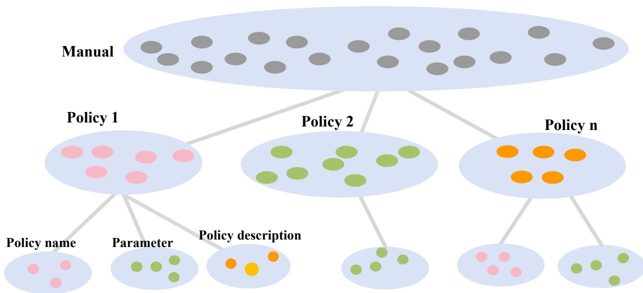


Fig. 5. Hierarchical knowledge base based on hardware manuals.

inherently possess a hierarchical text structure, we constructed the hierarchical knowledge base through rules or word vector clustering. The hierarchical knowledge base built based on hardware manuals is shown in Fig. 5. During the retrieval phase, we trained an index generator for efficient retrieval, with the implementation method referring to [45].

2) Metric-level PVG based on MoE: The primary aim of IBN is to satisfy the service requirements of users. However, given the typically large user base in IBN, it would be a significant waste of resources to train an individual optimization model for each user. Therefore, this paper proposes the application of MoE for generating metric-level PVs. This approach coordinates multiple expert models with diverse optimization objectives to address complex optimization tasks. Given the superior semantic understanding capabilities of LLM, they can be employed to replace the conventional Gate Network in MoE to generate the expert weight matrix. LLM is particularly effective in identifying implicit demands within intentions and coordinating expert models for efficient decision-making.

A set of expert models is trained through the Deep Reinforcement Learning (DRL) method. The objective of these experts is to maximize the benefits of each QoS indicator by selecting the optimal bandwidth. The network primarily considers three QoS indicators: Data Transfer Rate (DTR), Packet Loss Rate (PLR), and Bit Error Rate (BER), which are the three key indicators of QoS in data transmission. The main workflow of the metric-level PVG based on MoE is illustrated in Fig. 6. This method employs LLM to analyze intent and generates a weight matrix in conjunction with the optimization objectives of expert models. In the prompt text, text with underline should be pre-set according to different parameters.

In network scenarios, users often require high DTR to ensure rapid loading of high definition videos, low PLR to minimize live video stuttering, and low BER to decrease the likelihood of video and audio desynchronization. The Shannon formula is employed to calculate the DTR for each link, with the average value of the link serving as the indicator value for computing QoS benefits. The following section outlines the methodology employed to calculate the benefits of DTR:

$$\mathcal{T}_{DTR} = \frac{1}{L_{n_u,d_u}} \sum_{l\in L_{n_u,d_u}} B_u \log_2{(1+\frac{P_l}{\sigma_l})} \tag{5}$$

where P_l is the signal power of the link l, and σ_l is the noise power of the link l.

The calculation of the two QoS indicators, PLR and BER, is simulated by taking into account a range of influencing factors. The upper incomplete gamma function is utilized to simulate the PLR in response to changes in bandwidth, noise power, and the number of links. The following section outlines the methodology employed to calculate the benefits of PLR:

$$\mathcal{T}_{PLR} = 1 - \frac{1}{L_{n_u,d_u}} \sum_{l\in L_{n_u,d_u}} \frac{\Gamma(L, \frac{B_u}{\sigma^2})}{\Gamma(L_{n_u,d_u})} \tag{6}$$

where $\Gamma(\cdot)$ represents the upper incomplete gamma function. In order to calculate the gain of BER, the change in BER is simulated in relation to bandwidth, signal-to-noise ratio, and the maximum error rate g of a single bit. The following outlines the method for calculating the benefits of BER:

$$\mathcal{T}_{BER} = 1 - \frac{1}{L_{n_u,d_u}} \sum_{l\in L_{n_u,d_u}} (1-g)^{\overline{B_u \log_2{(1+\frac{P_l}{\sigma_l})}}^{\vartheta}} \tag{7}$$

where ϑ is an empirical parameter, obtained by simulating the change in BER. The application of the aforementioned three formulas in isolation as reward functions for the DRL model enables the effective training of an expert that maximises the gain of a specified QoS indicator.

In this paper, Deep Q-Network (DQN) [46] is employed as an expert model. A Markov decision process is constructed in which the network makes bandwidth allocation decisions based on the current network state. Upon the arrival of the next user request, the resulting network state depends solely on the previous state and action, adhering to the Markov property. The state space is defined by the remaining bandwidth resources B_l in link l, the source node n_u and destination node d_u of the service request, denoted as $S_u = \{n_u, d_u, B_{l_1}, B_{l_2}, ..., B_{l_{|L_{n_u,d_u}|}}\}$. The action space, $A = B_k = \{b_1, b_2, ..., b_k\}$, is defined as the set of bandwidths that the links of network can set. The rewards, denoted by $r_u = R_{u,t_n}$, are the gains of each QoS indicator obtained from the interaction between the model and the environment. These rewards are also subject to the constraint of Eq. (4d).

To facilitate comprehensive network state and edge space exploration and enhance model learning stability, a noise layer is integrated into DQN, replacing the conventional ϵ-greedy strategy [47]. The parameters of the behaviour network and target network are augmented with noise random variables ϵ and ϵ^-, as well as noise parameters σ and σ^-. Consequently, the objective function formula of the DQN is as follows:

$$\begin{aligned} L(\theta) = & E_{\epsilon^-,\epsilon}[E_{(S_u,a_u,r_u,S_{u+1})\sim D}[r_u + \\ & \gamma \max_{a*\in A} Q(S_{u+1}, a^*, \epsilon^-; \theta^-, \sigma^-) \\ & - Q(S_u, a_u, \epsilon; \theta, \sigma)]^2] \end{aligned} \tag{8}$$

where θ represents the model parameters of the behavior network, while θ^- represents the model parameters of the target network. S_{u+1} is the new state of the network. a_u is a possible action in the current state, and a^* is a possible action

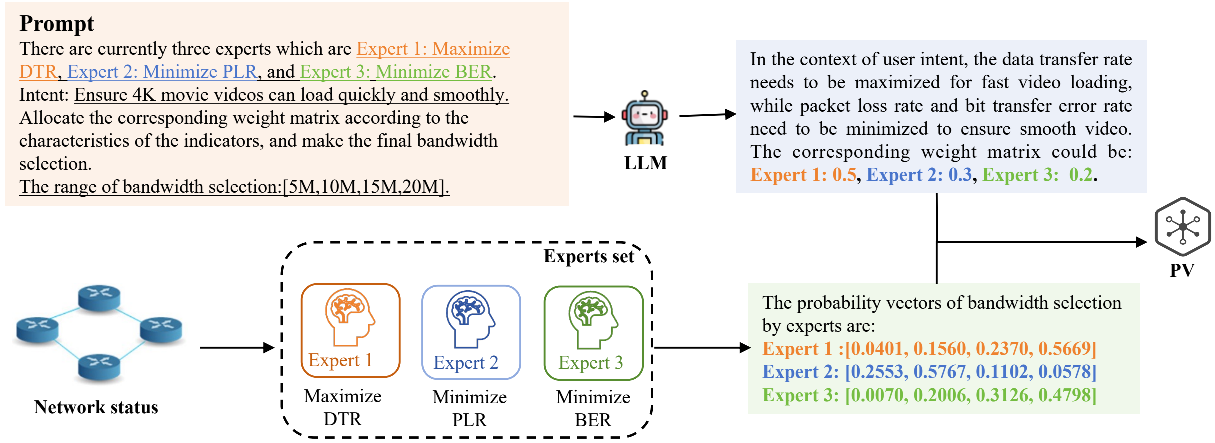


Fig. 6. The workflow of the metric-level PVG based on MoE. The Prompt consists of the expert's optimization directions, intent, and required operations. Based on the received Prompt, the LLM outputs an expert's weight matrix, which is then multiplied with the expert's results to generate the PV.

Algorithm 1: Expert Training Based On DQN

Input: Initial the network state s and $Q(s, a, \theta, \sigma, \epsilon)$
Output: Trained DQN model $Q(s, a, \theta, \sigma, \epsilon)$

1 Initialize $\hat{Q}(s, a, \theta^-, \sigma^-, \epsilon^-)$ with $\theta^- = \theta$, $\sigma^- = \sigma$ and $\epsilon^- = \epsilon$;
2 Initialize replay memory D to capacity N;
3 **for** *episode* = $1, ..., K$ **do**
4 $\quad s_u \leftarrow B_l, n_u$ and d_u;
5 $\quad$ **for** $u = 1, ..., U$ **do**
6 $\quad\quad a_u \leftarrow$ Get the bandwidth, $\arg\max_a Q(s_u, a, \theta, \sigma, \epsilon)$;
7 $\quad\quad$ Allocate the bandwidth a_u to the network;
8 $\quad\quad r_u \leftarrow$ Calculate the reward based on a_u;
9 $\quad\quad s_{u+1} \leftarrow$ get the new state based on a_u;
10 $\quad\quad$ Store transition (s_u, a_u, r_u, s_{u+1}) in D;
11 $\quad\quad$ Sample random minibatch of transitions (s_j, a_j, r_j, s_{j+1}) from D;
12 $\quad\quad y_j \leftarrow r_j + \gamma \max_{a^*} \hat{Q}(s_{j+1}, a^*, \theta^-, \sigma^-, \epsilon^-)$;
13 $\quad\quad$ Perform a gradient descent step on $(y_j - Q(s_j, a_j, \theta, \sigma, \epsilon))^2$ with respect to the weights θ;
14 $\quad\quad s_u \leftarrow s_{u+1}$;
15 $\quad\quad$ Every C steps reset $\hat{Q} \leftarrow Q$;

in the new state. γ is the discount factor. The training process of the expert based on DQN is elaborated in Algorithm 1.

The experience replay memory and the target action value function $\hat{Q}$ are initialized (Line 1-2). For each episode, the network is reinitialized and the current state is obtained. This includes the bandwidth resources of each link B_l, the source node n_u, and the target node d_u (Line 4). For each step, the model selects and executes an action, which is the bandwidth allocated to the service. It then calculates the reward, obtains the next new network state, and stores the transformation (Line 6-10). Subsequently, a small batch of transformations is randomly extracted from the memory and the target value is set (Line 11-12). The training step performs a gradient descent step with the objective of minimising the discrepancy between the prediction and the target and changing the state to the new one (Line 13-14). The target action value function is updated at regular intervals, with a specific number of steps being completed between each update (Line 15).

These models collectively constitute a set of expert models. With the analytical capacity of LLM, the requisite expert models can be derived directly from the user requirements and the corresponding weight matrix $W = \{w_1, w_2, ..., w_n\}$ can be attributed to the selected expert models. Each expert model generates a corresponding Q value vector based on the input network state, $Q(S_u, a_u) = \{q_1, q_2, ..., q_k\}$. The final weighted Q value vector can be represented as:

$$Q^*(S_u, a_u) = W \cdot \begin{bmatrix} Q_1(S_u, a_u) \\ Q_2(S_u, a_u) \\ \vdots \\ Q_n(S_u, a_u) \end{bmatrix} \tag{9}$$

The final selected action a^* can be represented as:

$$a^* = \arg\max_{a_u} Q^*(S_u, a_u) \tag{10}$$

The variable a^* represents the value of the metric-level parameter for bandwidth. This approach is applicable to other metric-level PVG tasks.

V. EVALUATION

A. Experiment Setup

Simulated network setting We use real-world network topologies, i.e., INS IXC Services and Uni-C Networks [48]. INS IXC Services is a topology with 33 nodes and 41 links, and Uni-C Networks has 54 nodes and 56 links. In the INS IXC Services, we set up 10 user nodes. Also, in the IRIS Networks, the number of user nodes are 20, respectively. The

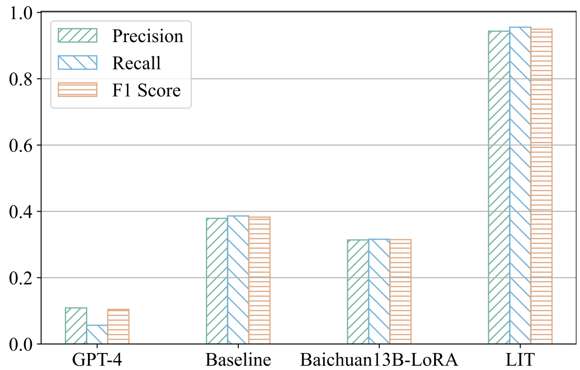


Fig. 7. The comparison of the Precision, Recall, and F1 Score results for intent translation between GPT-4, ChatGLM-6B (baseline), Baichuan-13B-LoRA, and LIT.

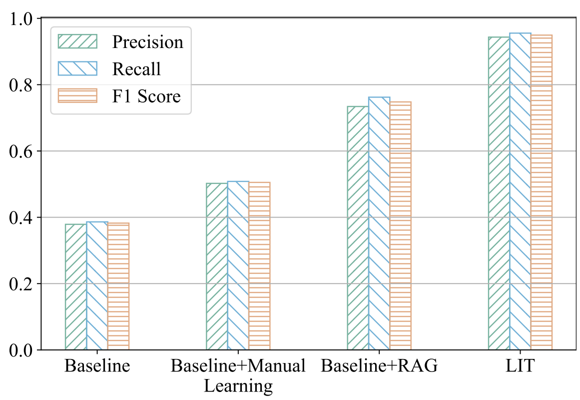


Fig. 8. The comparison of the Precision, Recall, and F1 score of the intent translation task with the introduction of manual learning and RAG.

initial bandwidth, signal power, and noise power for each link are set at 100 Mbps, 1000 W, and 1 W, respectively. A new parameter, ξ, is introduced to represent the network resource availability coefficient ranging from 0.3 to 1. For example, when $\xi = 0.5$, the maximum capacity of links is 50 Mbps, while when $\xi = 1.0$, the maximum capacity of links is 100 Mbps.

Relevant parameter setting In order to facilitate the translation of intent, we have collated open-source network management commands and manuals to serve as datasets and knowledge base. The collected data comprises 1,500 mappings of natural language intents to policy sets, with 90% designated for PSG training. Additionally, 500 policy and parameter explanation sets and 450 sets of relationships between policies and parameters are used as the dataset for Manual Learning. In PSG, we employ ChatGLM-6B [49] for full-parameter fine-tuning training, utilizing the Adam optimizer with the learning rate set to 10^{-4}. In PVG, ChatGLM-6B is also used for generating PVs. For MoE, the parameters are set as follows: $\beta_1 = 1$, $\beta_2 = 0.25$, $\alpha_1 = \alpha_2 = 0.5$, $r = 10$, $\vartheta = 5000$, and $g = 0.01$. The DQN models are trained by using the Adam

TABLE II
RESULTS OF ABLATION EXPERIMENTS FOR MANUAL LEARNING

Model	Metric			
	BLEU	Rouge-1	Rouge-2	Rouge-L
Baseline	62.20%	73.93%	71.64%	76.44%
Baseline+Manual Learning	96.33%	97.48%	97.29%	97.74%

optimizer with a learning rate of 10^{-3} and $\gamma = 0.1$, a batch size of 64, for 300 epochs. The interval for copying parameters between the estimated Q network and the target Q network is set to 1000. The nodes for the user and server are selected based on the user's intent, and the shortest path is calculated to determine the base stations traversed between them.

Evaluation metrics In order to assess the effectiveness of PSG, two NLP evaluation metrics, BLEU [50] and ROUGE [51], are employed. Additionally, ROUGE is comprised of three sub-metrics, namely ROUGE-1, ROUGE-2, and ROUGE-L. The correctness of LLM's intent translation is verified by comparing the final commands and PVs it generates. The evaluation considers three verification metrics: Recall, Precision, and F1 score. Recall is calculated as $\frac{M_r}{M}$, where M denotes the total number of parameters in the test set, and M_r denotes the parameters that fully match the ground truth. Precision is calculated as $\frac{M_r}{M_p}$, where M_p denotes the total number of parameters generated by LLM. The F1 score is calculated as $\frac{2 \cdot R \cdot P}{(R+P)}$. For the metric-level PV, the total QoS revenue, which are the sum up of the benefits of three QoS indicators, refer to Eq. (2), and the network total revenue composed of SSR rewards and QoS benefits, refer to Eq. (4a), are the most important metrics to evaluate the performance of the MoE method.

B. Performance of LIT

To demonstrate the advantages of LIT in intent translation, we conducted a comparative analysis of three representative LLMs (e.g., ChatGLM-6B, Baichuan-13B-LoRA [52], [53], and GPT-4) on this task in terms of Precision, Recall, and F1 score. We trained ChatGLM using full-parameter fine-tuning on the task, utilizing PSG and PVG for the training process. The fine-tuned model was then established as the baseline. For Baichuan, we applied LoRA training based on PSG and PVG. Due to the lack of an interface for full-parameter fine-tuning, GPT-4 was trained through manual learning via prompt engineering. The final results, as illustrated in Fig. 7, show that LIT possesses a strong capability for intent translation, achieving significant improvements over the other models. GPT-4, constrained by its dearth of support for comprehensive parameter fine-tuning and its exclusive reliance on manual learning, exhibited comparatively inferior performance. Baichuan, despite having achieved convergence in its loss function, did not perform as well as ChatGLM due to the limited number of parameters that could be adjusted

TABLE III
THE EXAMPLE OF COMPARING OUTPUTS FROM DIFFERENT SOLUTIONS

User Intent: Add a new base station for university laboratories, the main services are video calling and online meeting.
LIT: ADD SECTOR: SECTORID=1, ANTNUM=4, CREATESECTOREQM=TRUE; ADD CELL: LocalCellId=0, CellName='university laboratories', SECTORID=1, UlBandWidth=CELL_BW_N100, DlBandWidth=CELL_BW_N100, CustomizedULBandWidth=183, CustomizedDLBandWidth=183, UePowerMax=30; ACT CELL: LocalCellId=0;
Baseline+RAG: ADD CELL: LocalCellId=0, CellName='university laboratories', UlBandWidth=CELL_BW_N100, DlBandWidth=CELL_BW_N100, CustomizedULBandWidth=183, CustomizedDLBandWidth=183, UePowerMax=30; **Two policies are missing.**
Baseline+Manual learning: ADD SECTOR: SECTORID=1, ANTNUM=1, CREATESECTOREQM=TRUE; ADD CELL: LocalCellId=0, CellName='university laboratories', SECTORID=1, UlBandWidth=20, DlBandWidth=40, CustomizedULBandWidth=10, CustomizedDLBandWidth=10, UePowerMax=1; ACT CELL: LocalCellId=0;
Baseline: ADD CELL: LocalCellId=0, CellName='university laboratories', SECTORID=1, UlBandWidth=20, DlBandWidth=40, CustomizedULBand-Width=10, CustomizedDLBandWidth=10, UePowerMax=30; **Two policies are missing.**
GPT-4: CREATE CELL: CellId=0, CellName='university laboratories', BandWidth=20, latency=1; ACTIVE CELL: CellId=0;

through LoRA. Baseline faced problems such as grammatical errors and invalid PVs, due to not learning manual knowledge and failing to analyze service in the intent.

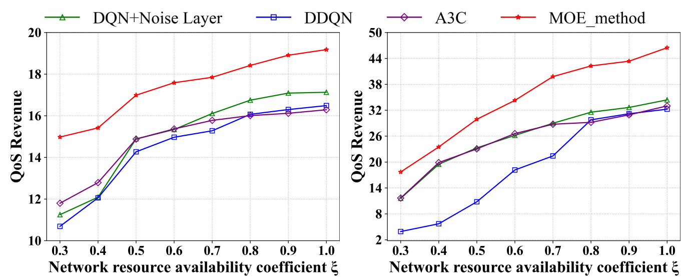
C. Manual Learning and RAG Performance

We design ablation experiments to demonstrate the effectiveness of manual learning in PSG and RAG in PVG, with results shown in Fig. 8. After incorporating manual learning, the F1 improved by 12.2% compared to the baseline. To further prove the effectiveness of manual learning, we evaluated the performance of the baseline and LIT in the PSG phase using different indicators, with results shown in Table II. After introducing RAG, the BLEU decreased by 34.13%, while ROUGE-L improved by 21.3%. Additionally, after introducing RAG, Precision, Recall, and F1 score improved compared to the baseline by 35.54%, 37.61%, and 36.55%, respectively.

An example of comparing outputs from different solutions is shown in Table III, where discrepancies in the outputs are highlighted in red font. LIT's output matches the true labels exactly. The output from Baseline+RAG lacks two policies, 'ADD SECTOR' and 'ACT CELL'. The output from Baseline+Manual learning contains some PV errors, such as 'ANTNUM=1' and others. The baseline can generate pilicies that generally comply with the manual's requirements, but it lacks some policies and parameters. Due to having undergone only instruction, GPT-4 produces strategies of lower quality.

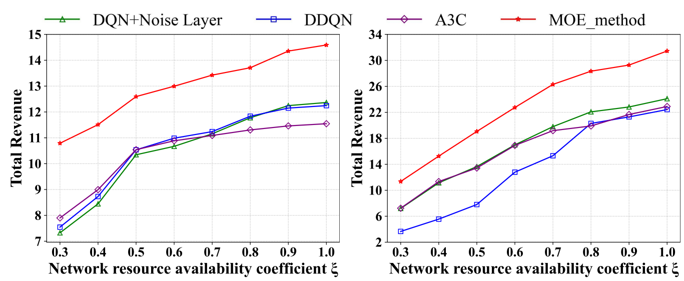
Based on the analysis of the above experimental results, we can draw the following conclusions:

- The performance of LIT surpasses other end-to-end methods in intent translation task. In the process of PSG, LIT ensures that the PS is syntactically correct and logical, laying the foundation for overall correctness. LIT generates the PVs more precisely, ultimately producing a high-quality policy set.
- Lacking manual learning leads to difficulties for the model in generating unknown policies during the PSG phase. This suggests that manual learning can enhance PSG performance by enabling the model to draw insights from the corpus of manuals, thereby reducing error propagation.
- The absence of RAG frequently results in the generation of invalid PVs during the PVG phase, causing policies to be undeployable. We attribute this to the inherent illusions problem of LLMs. Moreover, the lack of manual learning also affects PVG.



(a) INS IXC service (b) Uni-C network

Fig. 9. The results of QoS revenue under different network topologies.

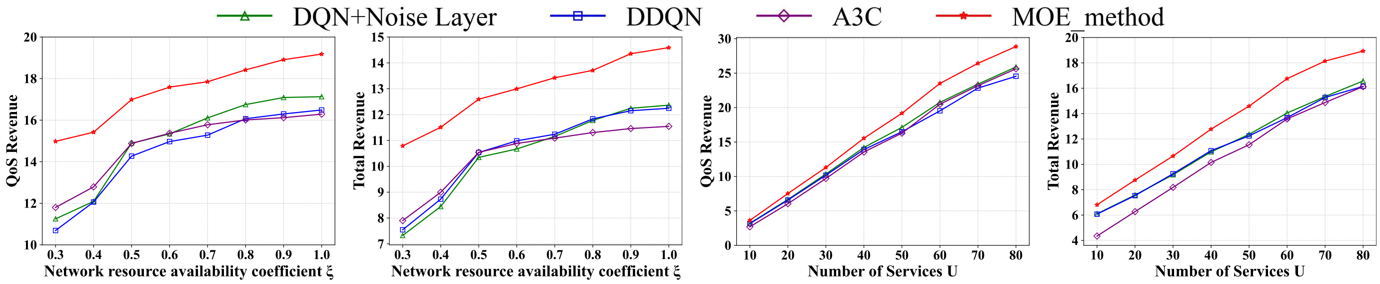


(a) INS IXC service (b) Uni-C network

Fig. 10. The results of total revenue under different network topologies.

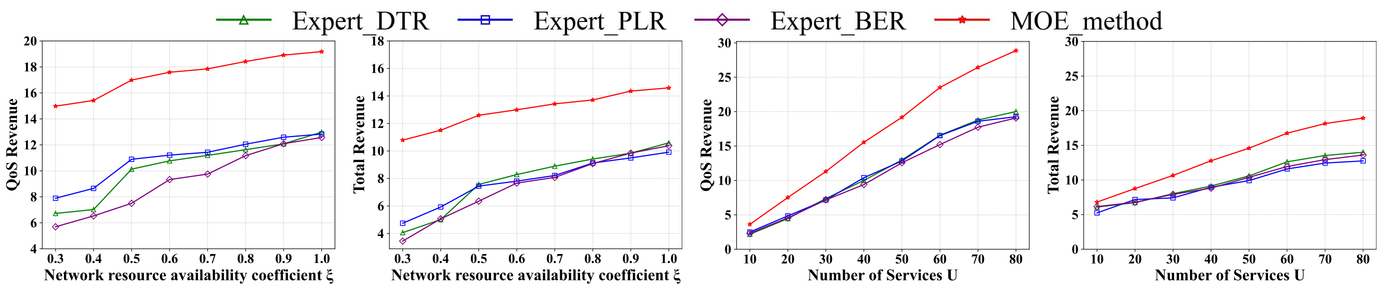
D. MoE Performance

The introduction of MoE significantly mitigated mismatches between user demands and network status, reducing the like-



(a) QoS revenue (b) Total revenue (c) QoS revenue (d) Total revenue

Fig. 11. The performance of different methods under varying network resource availability coefficients ξ and different numbers of services U.



(a) QoS revenue (b) Total revenue (c) QoS revenue (d) Total revenue

Fig. 12. The performance of MoE approach compared to single expert models under varying network resource availability coefficients ξ and different numbers of services U

lihood of resource conflicts. We compared the impact of MoE with that of three typical DRL models on total QoS and total revenue under different network topology scale, as shown in Fig. 9 and Fig. 10. The inputs and outputs of DQN, DDQN, and A3C are identical to expert models. Analysis of the results showed that, under high network resource availability coefficient (ξ=0.3), Moe improved QoS revenue by 26.9% and 52.67% in INS IXC Services and Uni-C network respectively, as shown in Fig. 9(a) and Fig. 9(b). The total revenue were improved by 36.6% and 56.95% in both network, as shown in Fig. 10(a) and Fig. 10(b). These experimental results prove that LLMs, by understanding user intents, can accurately adjust the weights of the results from various expert model under different network topology scale.

To further substantiate this conclusion, we compared the the performance of different methods under varying network resource availability coefficients ξ and different numbers of services in the network of INS IXC Services, as shown in Fig. 11. When the number of services increased to 80, MoE improved the QoS revenue and total revenue by 11.5% and 14.2%, respectively, as shown in Fig. 11(c) and Fig. 11(d). We also compared the performance of MoE and single expert models, as shown in Fig. 12. A single expert model could achieve locally optimal results when resources are not tight or the number of users was low. However, when network resources are scarce or the number of users increases, MoE can achieve better revenue.

VI. CONCLUSION

This paper proposes an LLM-empowered wireless network intent translation framework, named LIT. LLMs, known for their excellent semantic understanding and generation capabilities, are particularly well-suited for intent translation tasks. To more effectively address the challenges of hardware heterogeneity and highly dynamic networks faced by IBN, new components are introduced. The incorporation of RAG addresses the issue of LLMs generating illegal parameter values. Furthermore, the integration of MoE effectively aligns user demands with downstream network resources. Compared to the baseline, the F1 score improved by 56.7%. Extensive experiments demonstrate that LIT deeply analyzes user intents and translates them into high-quality policies, alleviating the challenges faced by IBN. In future work, we intend to apply LIT to large-scale industrial networks and enhance it with features such as detection and self-healing in order to establish a closed-loop IBN system.

VII. ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grants (U23B2001,62171057, 62101064, 62201072, 62001054, 62071067), the Ministry of Education and China Mobile Joint Fund (MCM20200202, MCM20180101), Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center.

REFERENCES

[1] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6g be?" Nature Electronics, vol. 3, no. 1, pp. 20–29, 2020.

[2] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6g wireless networks: Vision, requirements, architecture, and key technologies," IEEE vehicular technology magazine, vol. 14, no. 3, pp. 28–41, 2019.

[3] A. Leivadeas and M. Falkner, "A survey on intent-based networking," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, 2022.

[4] C. De Alwis, P. Kumar, Q.-V. Pham, K. Dev, A. Kalla, M. Liyanage, and W.-J. Hwang, "Towards 6g: Key technological directions," ICT Express, vol. 9, no. 4, pp. 525–533, 2023.

[5] M. Beshley, N. Kryvinska, and H. Beshley, "Quality of service management method in a heterogeneous wireless network using big data technology and mobile qoe application," Simulation Modelling Practice and Theory, vol. 127, p. 102771, 2023.

[6] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," Multimedia tools and applications, vol. 82, no. 3, pp. 3713–3744, 2023.

[7] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," arXiv preprint arXiv:1506.00019, 2015. [Online]. Available: http://arxiv.org/abs/1506.00019

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, jun 2019, pp. 4171–4186.

[9] J. Mcnamara, D. Camps-Mur, M. Goodarzi, H. Frank, L. Chinchilla-Romero, F. Cañellas, A. Fernández-Fernández, and S. Yan, "Nlp powered intent based network management for private 5g networks," IEEE Access, 2023.

[10] A. Banerjee, S. S. Mwanje, and G. Carle, "An intent-driven orchestration of cognitive autonomous networks for ran management," in 2021 17th International Conference on Network and Service Management (CNSM). IEEE, 2021, pp. 380–384.

[11] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," IEEE Access, vol. 8, pp. 22 862–22 873, 2020.

[12] C. Yang, X. Mi, Y. Ouyang, R. Dong, J. Guo, and M. Guizani, "Smart intent-driven network management," IEEE Communications Magazine, vol. 61, no. 1, pp. 106–112, 2023.

[13] A. S. Jacobs, R. J. Pfitscher, R. H. Ribeiro, R. A. Ferreira, L. Z. Granville, W. Willinger, and S. G. Rao, "Hey, lumi! using natural language for intent-based network management," in 2021 USENIX Annual Technical Conference (USENIX ATC 21), 2021, pp. 625–639. [Online]. Available: https://www.usenix.org/conference/atc21/presentat-ion/jacobs

[14] A. S. Jacobs, R. J. Pfitscher, R. A. Ferreira, and L. Z. Granville, "Refining network intents for self-driving networks," in Proceedings of the Afternoon Workshop on Self-Driving Networks, 2018, pp. 15–21.

[15] J. Zhao, H. Sun, J. Wang, Q. Qi, Z. Zhuang, S. Tao, and J. Liao, "Confpilot: A pilot for faster configuration by learning from device manuals," in 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). IEEE, 2023, pp. 108–119.

[16] H. Chen, Y. Miao, L. Chen, H. Sun, H. Xu, L. Liu, G. Zhang, and W. Wang, "Software-defined network assimilation: bridging the last mile towards cen- tralized network configuration management with nassim," in Proceedings of the ACM SIGCOMM 2022 Conference, 2022, pp. 281–297.

[17] C. Ge, Z. Ge, X. Liu, A. Mahimkar, Y. Shaqalle, Y. Xiang, and S. Pathak, "Chroma: Learning and using network contexts to reinforce performance improving configurations," in Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, 2023, pp. 1–16.

[18] R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do llms need to synthesize correct router configurations?" in Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, 2023, pp. 189–195.

[19] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice lifecycle management for 5g mobile networks: An intent-based networking approach," IEEE Access, vol. 9, pp. 80 128–80 146, 2021.

[20] Z. Chu, L. Guo, J. Wang, Q. Qi, Z. Zhuang, H. Sun, and C. Zhou, "Hirn: A hierarchical intent refinement approach for dependable network slicing with multi-path resource allocation," in 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). IEEE, 2023, pp. 7–13.

[21] I. Ahmad, J. Malinen, F. Christou, P. Porambage, A. Kirst"adter, and J. Suomalainen, "Security in intent-based networking: Challenges and solutions," pp. 1–8, 2023.

[22] M. Sophocleous, C. Lessi, Z. Xu, J. Špaňhel, R. Qiu, A. Lendinez, I. Chondroulis, and I. Belikaidis, "Ai-driven intent-based networking for 5g enhanced robot autonomy," in IFIP International Conference on Artificial Intelligence Applications and Innovations, ser. IFIP Advances in Information and Communication Technology, vol. 652. Springer, 2022, pp. 61–70.

[23] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[24] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," ACM Computing Surveys, vol. 56, no. 2, pp. 1–40, 2023.

[25] L. Chen, M. Zaharia, and J. Zou, "How is chatgpt's behavior changing over time?" Harvard Data Science Review, vol. 6, no. 2, mar 12 2024. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/y95zitmz

[26] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," IEEE Transactions on Knowledge and Data Engineering, 2023.

[27] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, 2023.

[28] B. Lin, D. Bouneffouf, G. A. Cecchi, and K. R. Varshney, "Towards healthy AI: large language models need therapists too," CoRR, vol. abs/2304.00416, 2023.

[29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, and T. Rocktäschel et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[30] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," CoRR, vol. abs/2312.10997, 2023.

[31] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," CoRR, vol. abs/2402.19473, 2024.

[32] Y. Lou, F. Xue, Z. Zheng, and Y. You, "Sparse-mlp: A fully-mlp architecture with conditional computation," arXiv preprint arXiv:2109.02008, vol. 1, p. 12, 2021. [Online]. Available: https://arxiv.org/abs/2109.02008

[33] S. Shen, Z. Yao, C. Li, T. Darrell, K. Keutzer, and Y. He, "Scaling vision-language models with sparse mixture of experts," arXiv preprint arXiv:2303.07226, 2023.

[34] H. Du, G. Liu, Y. Lin, D. Niyato, J. Kang, Z. Xiong, and D. I. Kim, "Mixture of experts for network optimization: A large language model-enabled approach," arXiv preprint arXiv:2402.09756, 2024.

[35] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat et al., "Glam: Efficient scaling of language models with mixture-of-experts," in International Conference on Machine Learning. PMLR, 2022, pp. 5547–5569. [Online]. Available: https://proceedings.mlr.press/v162/du22c.html

[36] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," arXiv preprint arXiv:1701.06538, 2017. [Online]. Available: http://arxiv.org/abs/1701.06538

[37] H. Hazimeh, Z. Zhao, A. Chowdhery, M. Sathiamoorthy, Y. Chen, R. Mazumder, L. Hong, and E. Chi, "Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning," Advances in Neural Information Processing Systems, vol. 34, pp. 29 335–29 347, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/f5ac21cd0ef1b88e9848571aeb53551a-Abstract.html

[38] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. V. Le, J. Laudon et al., "Mixture-of-experts with expert choice routing," Advances in Neural Information Processing Systems, vol. 35, pp. 7103–7114, 2022. [Online].

Available: http://papers.nips.cc/paper_files/paper/2022/hash/2f00ecd787b432c1d36f3de9800728eb-Abstract-Conference.html
[39] Y. Wei, M. Peng, and Y. Liu, “Intent-based networks for 6g: Insights and challenges,” Digital Communications and Networks, vol. 6, no. 3, pp. 270–280, 2020.
[40] Y. Abassi and N. Tabbane, “Context aware life cycle for intent-based networking systems,” in 2023 IEEE Tenth International Conference on Communications and Networking (ComNet). IEEE, 2023, pp. 1–8.
[41] A. Rafiq, A. Mehmood, T. Ahmed Khan, K. Abbas, M. Afaq, and W.-C. Song, “Intent-based end-to-end network service orchestration system for multi-platforms,” Sustainability, vol. 12, no. 7, 2020.
[42] H. Xu, B. Liu, L. Shu, and P. S. Yu, “Bert post-training for review reading comprehension and aspect-based sentiment analysis,” arXiv preprint arXiv:1904.02232, 2019.
[43] S. Hameche, M. E. Khanouche, A. Chibani, and A. Tari, “A group teaching optimization-based approach for energy and qos-aware internet of things services composition,” Journal of Network and Systems Management, vol. 32, no. 1, p. 4, 2024.
[44] I. Chakour, C. Daoui, M. Baslam, B. Sainz-de Abajo, and B. Garcia-Zapirain, “Strategic bandwidth allocation for qos in iot gateway: Predicting future needs based on iot device habits,” IEEE Access, vol. 12, pp. 6590–6603, 2024.
[45] Y. Tay, V. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. Gupta et al., “Transformer memory as a differentiable search index,” Advances in Neural Information Processing Systems, vol. 35, pp. 21 831–21 843, 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/892840a6123b5ec99ebaab8be1530fba-Abstract-Conference.html
[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” arXiv preprint arXiv:1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602
[47] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, “Noisy networks for exploration,” CoRR, vol. abs/1706.10295, 2017. [Online]. Available: http://arxiv.org/abs/1706.10295
[48] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The internet topology zoo,” IEEE Journal on Selected Areas in Communications, vol. 29, no. 9, pp. 1765–1775, 2011.
[49] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang, “Glm: General language model pretraining with autoregressive blank infilling,” arXiv preprint arXiv:2103.10360, 2021.
[50] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[51] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in Text summarization branches out, 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013
[52] A. Yang, B. Xiao, B. Wang, B. Zhang, C. Bian, C. Yin, C. Lv, D. Pan, D. Wang, D. Yan et al., “Baichuan 2: Open large-scale language models,” arXiv preprint arXiv:2309.10305, 2023.
[53] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” in ICLR 2022, April 2022. [Online]. Available: https://www.microsoft.com/en-us/research/publication/lora-low-rank-adaptation-of-large-language-models/