Proactive Intent-Driven SFC Placement for Dynamic IoT Networks

Zijie Huang, *Member, IEEE,* Lizhao Wu*, and Hui Lin*, *Senior Member, IEEE*

***Abstract*—The proliferation of Internet of Things (IoT) devices necessitates highly flexible and efficient network infrastructures, with Service Function Chaining (SFC) playing a crucial role in delivering customized network services. While Intent-Based Networking (IBN) simplifies network management by translating high-level intents into configurations for SFC placement, current state-of-the-art solutions are predominantly reactive. They struggle with the unpredictable dynamism and resource heterogeneity of IoT traffic, the semantic gap in intent translation, and a fundamental lack of proactive adaptation, leading to suboptimal resource utilization and compromised Quality of Service. To address these critical challenges, this paper proposes a novel proactive intent-driven SFC placement framework for dynamic IoT networks. Our solution synergistically integrates a Transformer-based module for accurately predicting future network intents and resource demands, an LLM-driven module for robustly translating complex natural language intents into precise network configurations, and a Reinforcement Learning (RL) agent for adaptive and optimal SFC deployment. We aim to demonstrate the efficiency of our proposed framework through comprehensive evaluations focusing on SFC acceptance rate, long-term revenue-to-cost (LRC), and resource allocation efficiency, including ablation studies to quantify the individual contributions of the prediction and translation modules.**

***Index Terms*—SFC placement, IBN, IoT resource allocation, transformer, large language model**

I. Introduction

THE rapid proliferation of Internet of Things (IoT) devices across various domains, from smart cities and industrial automation to healthcare and connected vehicles, has fundamentally transformed network landscapes [1][2]. This pervasive connectivity generates unprecedented volumes of diverse data, necessitating highly flexible, scalable, and efficient network infrastructures [3]. Within this context, Service Function Chaining (SFC) has emerged as a crucial paradigm. SFC allows for the dynamic creation of ordered sequences of Virtual Network Functions (VNFs) (e.g., firewalls, load balancers, intrusion detection systems) that process network traffic to deliver customized services [4][5]. The efficient placement of these SFCs, which involves optimally mapping virtual functions and their interconnections onto the underlying physical network resources while satisfying various constraints (e.g., latency, bandwidth, CPU), is critical for ensuring the performance and reliability of these services. Complementing SFC, Intent-Based Networking (IBN) represents a significant evolution in network management [6]. IBN shifts the focus from low-level, imperative network configurations (specifying "how" to do something) to high-level, declarative business objectives or "intents" (specifying "what" needs to be achieved) [7]. By abstracting the underlying network complexities, IBN aims to simplify network operations, enhance automation, and align network behavior directly with business goals. Within the IoT context, where service requirements are highly variable and often domain-specific, an intent-driven approach offers a promising pathway to more context-aware and responsive SFC placement [8]. This paradigm shift toward declarative network management paves the way for integrating self-awareness into orchestration systems, thereby enhancing automation, adaptability, and efficiency in dynamic IoT networks.

Followed by the above technologies background, several approaches have been proposed recently to advance SFC placement efficiency with IBN. For instance, Leivadeas and Falkner [9] propose to translate network service requirements from the users via IBN into VNF deployment solutions. Their solution presents automatic configuration of the network service, and high quality of service and security requirements. Similarly, Avgeris et al. proposed an automatic orchestration of network services in IBN enabled SFC [10]. Their method dynamically reassigning incoming intents among the associated SFCs to proactively execute corrective actions. Their experimental results demonstrate that they can assure high application probability and minimize QoS violations.

However, a key limitation of this current state-of-the-art in IBN-enabled SFC placement is its predominantly reactive nature. While these systems automate deployment based on current intents, they generally lack sophisticated mechanisms to anticipate future network states, traffic fluctuations, or evolving user demands [11]. This reliance on reactive adjustments directly leads to several critical challenges, especially within dynamic IoT networks. Firstly, the unpredictable dynamism of IoT traffic, characterized by sporadic, bursty, and unpredictable patterns driven by events and device mobility, renders static or purely reactive SFC placement suboptimal, often resulting in inefficient resource utilization, increased latency, or service disruptions. Secondly, resource heterogeneity and constraints in IoT deployments, involving a vast array of devices and edge computing nodes with diverse and often limited resources, present a complex resource allocation problem that reactive approaches struggle to optimize effectively [12]. Thirdly, a persistent semantic gap in intent management remains, as

Zijie Huang is with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, 350117, China, and Faculty of Science and Engineering, School of Electrical, Electronic and Mechanical Engineering, University of Bristol, Bristol, BS8 1TR, United Kingdom (email: z.huang@bristol.ac.uk). Hui Lin, Lizhao Wu are with the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, 350117, China, Email: linhui@fjnu.edu.cn, melowlz@yeah.net.
Corresponding authors: Hui Lin, Lizhao Wu.

translating nuanced, high-level natural language intents (e.g., "ensure ultra-low latency for critical sensor data") into precise, executable SFC configurations is challenging due to ambiguity and context-dependency, limiting the full potential of intent-driven automation [13]. Consequently, the lack of proactive adaptation means that most existing systems merely react to network changes or new service requests, failing to optimize SFC placement in anticipation of future conditions, which leads to delays in service provisioning and suboptimal performance during peak loads or unexpected events. Finally, the scalability and real-time decision-making demands of the immense scale of IoT devices and the need for immediate responsiveness in dynamic environments necessitate placement solutions that can make rapid, intelligent decisions without prohibitive computational overhead, a requirement often unmet by current reactive frameworks.

To tackle the challenges outlined above, we propose a proactive intent-driven SFC placement solution, which integrates Transformer-based intent prediction, LLM-driven intent translation, and Reinforcement Learning (RL)-Agent for dynamic SFC deployment in IoT networks. This solution aims to achieve efficient and intelligent real-time decision-making of SFC placement in dynamic IoT networks. The main contributions of our work are are summarized as follows:

- To overcome the unpredictable dynamism of IoT traffic and the lack of proactive adaptation, we leverage Transformer to analyze IoT physical network infrastructure (e.g., topology, real-time available resources), and historical intent requests to accurately predict future network intents and resource demands. This proactive forecasting capability allows the system to anticipate changes and prepare the network for upcoming service requirements.
- To bridge the semantic gap in intent management, we employ Large Language Model (LLM) to bridge the semantic gap between human-expressed intents and machine-executable network configurations. The LLM component translates high-level natural language intents into detailed SFC policies and VNF parameters, handling contextual nuances and generating precise, structured deployment instructions.
- We demonstrate the efficiency of our proposed solution by evaluating it from three aspects: SFCs placement network performance, efficiency of LLM adoption for intent translation, and ablation study for transformer-based intent prediction.

The remainder of this paper is organized as follows. Section II presents related work. Section III formulates the system model and objective. In Section IV, we elaborate our framework. Experimental results and analysis are conducted in Section V. Finally, Section VI concludes this paper.

II. Background and Related Work

A. SFC Placement in Intent-Based Networking Frameworks

More and more recent works are aiming to automate the complex process of SFC deployment based on high-level intents. For instance, Chowdhury proposed an end-to-end network management and user intent-aware intelligent network resource-slicing scheme for SFC-based network application [14]. They considered different intents such as time-first and cost-first based on resource-slicing policies for Zero Touch Network-based 6G application. Their experimental results presented a 11.52% service monetary gain, and a 6.15% energy gain. Subsequently, Avgeris et al. proposed an automated network assurance model to guarantee the Quality of Service (QoS) and security requirements of IBN-enabled SFCs in IoT and 5G network service background [8]. A model predictive control-based algorithm is introduced to proactively and optimally assign the incoming intents among the available SFCs. Their experimental results demonstrated that their method can maximize the Service-Level Agreement satisfaction, and minimize QoS violations. Furthermore, to achieve intent-oriented applications in network slicing, Zou et al. proposed a hypergraph theory to realize the customization of network service application intents, which is able to link the application intents with network slicing strategies [15]. Their method presented significant enhancement on user acquisition precision, network resource optimization, and customization of slice generation improvement.

While these IBN-enabled SFC placement solutions offer improved automation and management simplicity compared to traditional methods, they predominantly remain reactive. They respond to new intents or detected network state deviations, lacking the inherent foresight to anticipate future demands or potential network issues. This reactive nature, particularly in the face of the unpredictable dynamism of IoT traffic, leads to suboptimal resource utilization and can hinder the achievement of stringent QoS requirements.

B. Advanced AI/ML in Network Management

The increasing complexity and dynamism of modern networks have driven significant interest in applying advanced Artificial Intelligence and Machine Learning techniques to various network management tasks. Reinforcement Learning (RL), in particular, has shown promise for dynamic resource allocation and SFC placement problems [16][17][18], as it allows agents to learn optimal decision-making policies through interaction with the network environment. Beyond deployment, AI (such as Transfer Learning, Graph Neural Network and Federated Learning and so on) has been applied to network monitoring, cyber security, and traffic classification [19][20][21][22][23][24]. More recently, LLMs have demonstrated remarkable capabilities in natural language understanding and generation, leading to their exploration in network management for tasks such as natural language-to-configuration translation. For instance, Tu et al. proposed an Network Function Virtualization (NFV)-intent for in-context learning in LLMs to perform the intent translation task [25]. Their experiment showed that the intent can be translated into JSON configuration with high accuracy. Additionally, Alam and Song also proposed an LLMs based-IBN to translate high-level user intents into actionable network policies for Space-Air-Ground Integrated Network (SAGIN) [26]. Their experimental results presented a latency reduction, bandwidth utilization improvement, QoS violations minimization, and

SFC acceptance rate improvement. Furthermore, Mekrache and Ksentini introduced an intent translation system based on LLMs to convert natural language intents into network service descriptors [27]. They trained an open-source LLM with few-shot examples from a knowledge base, which is being refined with a human feedback mechanism to improve the system's performance over time.

While these individual AI/ML techniques have significantly advanced specific aspects of network operations, their integrated and synergistic application, particularly for proactive intent-driven SFC placement in highly dynamic IoT environments, remains an underexplored area. Existing solutions often lack the comprehensive foresight to truly anticipate future demands and the flexible, human-centric translation capabilities needed for diverse IoT service requirements.

C. Gap and Our Contribution

The above existing literature has made significant strides in optimizing SFC placement and automating network management through IBN. However, a critical gap persists in the ability of current IBN-enabled SFC placement solutions to proactively adapt to the inherent dynamism of IoT networks. Specifically, the reactive nature of current systems, coupled with the challenges of accurately translating nuanced human intents and efficiently deploying SFCs in real-time within resource-constrained IoT environments, necessitates a more intelligent and anticipatory framework. Our work directly addresses this gap by proposing a novel framework that integrates Transformer-based intent prediction, LLM-driven intent translation, and RL-Agent for dynamic SFC deployment. This unique combination enables truly proactive network management, anticipating future demands and optimizing SFC placement to ensure superior performance and resource efficiency in dynamic IoT networks.

III. System Model & Problem Formulation

A. IoT Network Infrastructure

We consider a heterogeneous IoT network infrastructure represented as a graph $G^P = (N^P, L^P)$, where N^P is the set of physical nodes and L^P is the set of physical links connecting them. The physical nodes $n \in N^P$ represent diverse computing resources distributed across the network. Each physical node n is characterized by its available computational capacity, denotes as C_n^{CPU} (numbers of CPU cores), C_n^{Mem} (size of memory in GB), and C_n^{GPU} (numbers of GPU cores). Each physical link $l \in L^P$ is characterized by its available bandwidth capacity B_l. We assume that L^P includes both wired and wireless connections, reflecting the diverse connectivity options in IoT environments.

On top of the physical infrastructure, VNFs are deployed as software instances. Each VNF v_i has specific resource requirements, including $r_{v_i}^{CPU}$ (CPU demand), $r_{v_i}^{Mem}$ (memory demand), and $r_{v_i}^{GPU}$ (GPU demand). Multiple instances of the same resource required VNF can be deployed across different physical nodes, subject to their available capacities. Meanwhile, SFCs S_k are defined as an ordered sequence of these VNFs $v_i \in S_k$ that collectively provide a network service.

B. Objective Function

The primary objective of our proactive intent-driven SFC placement framework is to maximize the overall long-term value generated by placing SFC requests in dynamic IoT networks, while efficiently utilizing network resources. This involves optimizing for both current and predicted future intents. Specifically, for a given time horizon T, our goal is to: *Maximize Long-term Revenue-to-Cost (LRC) and SFC Request Acceptance Rate (RAC), while ensuring Resource Allocation Efficiency.* Let K be the set of all intents arriving or active within the time horizon T. For each intent $I_k \in K$, let $a_k \in 0, 1$ be a binary decision variable indicating whether SFC S_k derived from I_k is accepted and successfully placed ($a_k = 1$) or rejected ($a_k = 0$). The objective function can be formulated as:

$$\text{Maximize} \quad \mathcal{O} = \text{LRC} + \text{RAC} \tag{1}$$

in which the LRC can be represented as:

$$\text{RAC} = \frac{\sum_{t=0}^{T} \sum_{S_k \in S_K(t)} REV(S_k) \times \omega}{\sum_{t=0}^{T} \sum_{S_k \in S_K(t)} COST(S_k) \times \omega} \tag{2}$$

where $S_K(t)$ is the set of all SFC requests arriving at time slot t. ω is the lifetime of the current SFC request S_k. $REV(S_k)$ is the revenue generated by deploying this current SFC S_k. $COST(S_k)$ is the resource consumption in the physical network due to the deployment S_k, which is represented by the sum of the physical network node resources and physical bandwidth resources scaled by path length. and RAC is represented as:

$$\text{LRC} = \frac{\sum_{t=0}^{T} |S_k(t)|}{\sum_{t=0}^{T} |S_K(t)|} \times 100\% \tag{3}$$

The objective function is subjected to various constraints. Firstly, for every physical node $n \in N_P$, the total CPU, GPU, and memory demanded by all the VNFs placed on it must not excceed its capacity:

$$\sum_{v \in N_k^V \text{ placed on } n} r_v^{CPU,GPU,Mem} \leq C_n^{CPU,GPU,Mem} \, \forall n \in N_P \tag{4}$$

in the meanwhile, link capacity constraint also appears as node capacity constraints: for every physical link $l \in L_P$, the total bandwidth demanded y all virtual links mapped to it must not exceed its capacity:

$$\sum_{(v_i, v_j \in L_k^L) \text{ mapped to } l} b_{v_i, v_j}^{S_k} \leq B_l \, \forall l \in L_P \tag{5}$$

on top of the physical and node capacity constraints, the logical order of VNFs within an SFC must be preserved during mapping to physical paths.

IV. The Proposed Method

In this section, we demonstrate our proposed proactive intent-driven SFC placement framework for dynamic IoT networks. The proposed framework consists of three modules: a LLM-based intent translation module, the transformer-based intent prediction module, and the intent-based SFC placement

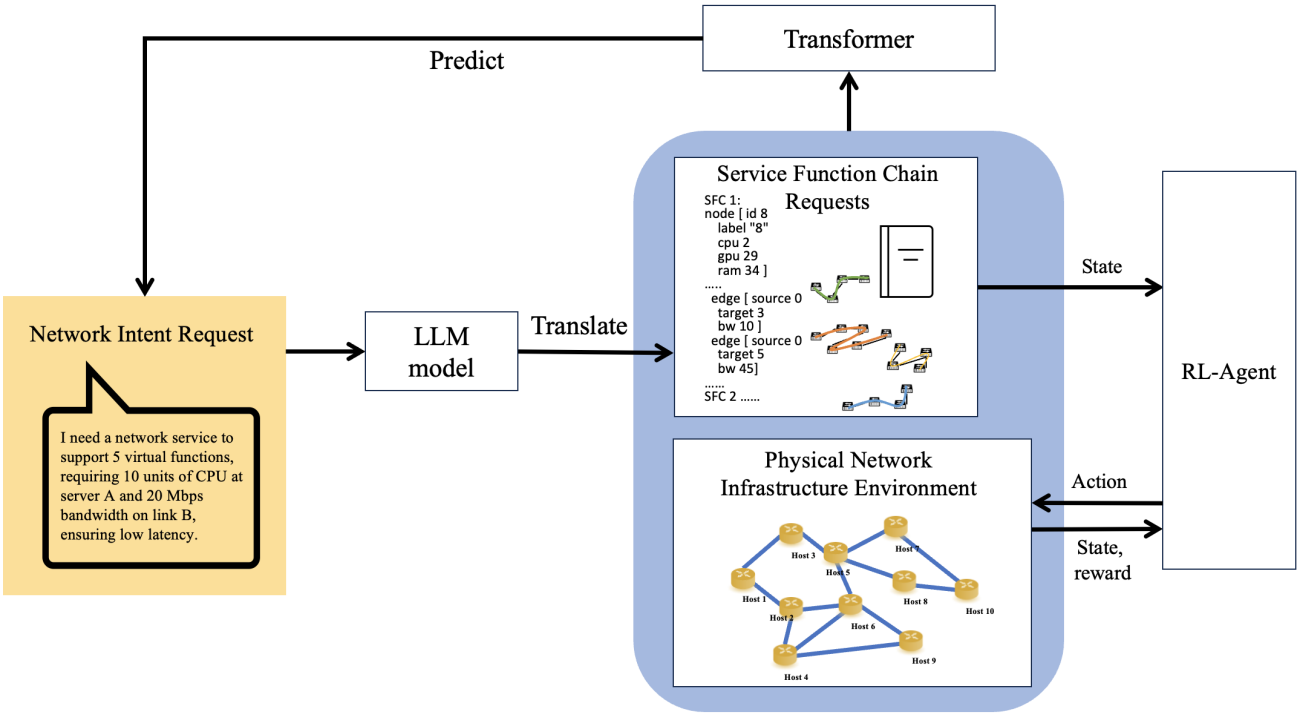

Fig. 1: Proactive Intent-Driven SFC Placement Framework

mechanism. In what follows we first provide an overview of the system architecture, followed by a comprehensive description of its key components.

A. Framework Overview

Our proposed framework operates as a closed-loop intelligent system designed to proactively manage and optimize SFC placement in dynamic IoT environments. As depicted in Fig. 1, the system comprises several interconnected modules that work synergistically to translate high-level user intents into actionable network configurations, anticipate future demands, and dynamically deploy SFCs. The workflow initiates with the ingestion of high-level human-expressed network intent request, such as "I need a network service to support 5 virtual functions, requiring 10 units of CPU at node A and 20 Mbps bandwidth on link B, ensuring low latency.", which defines the desired network service outcomes for IoT applications. This ingested intent is then processed by an LLM model, responsible for translating these natural language intents into structured "Virtual Network Setting", encompassing detailed VNF parameters and SFC graph manners (such as parallel or sequential structure). Concurrently, real-time physical network status, including physical network topology and available resources, alongside the translated intent data, are feeding into a RL-Agent. The RL-Agent, acting as the intelligent orchestrator, makes dynamic SFC placement decisions. A continuous feedback loop is established where real-time performance metrics and updated network state information from the NFV Environment are fed back to the RL-Agent for learning and policy refinement. Lastly, a crucial proactive element is introduced by feeding the current translated intents and physical network status will feed into a Transformer model to predict future intents. This module provides essential foresight improving the accuracy of subsequent intent predictions, thereby ensuring a truly adaptive and optimized closed-loop system.

B. LLM-based Intent Translation

The LLM-based Intent Translation module is responsible for bridging the semantic gap between human-centric language and machine-executable network configurations. Upon receiving a high-level intent, the LLM processes this unstructured natural language input to extract all relevant entities and parameters. This involves identifying the specific service functions required, their logical ordering, and the associated Quality of Service (QoS) constraints (e.g., latency thresholds, bandwidth guarantees).

The LLM is designed to perform this translation by leveraging its advanced understanding of context and its ability to generate structured output. It converts the qualitative and often ambiguous intent into a precise "Virtual Network Settings" object. This object includes: 1) A formal description of the SFC, specifying the sequence of VNF types; 2) Detailed resource requirements for each VNF instance (e.g., CPU, memory, GPU).

Our proposed LLM-based Intent Translation significantly enhances the usability and flexibility of the intent-driven system, moving beyond rigid template-based approaches by allowing operators to express complex requirements in a more intuitive manner.

TABLE I: Typical intents in NLU dataset

Intent	Category	Description	Network Entities
Create Flow	Simple	Create a network flow between two endpoints with optional constraints, such as CPU, GPU, MEM, BW.	Mandatory: (source, target) Optional: (CPU, GPU, MEM, BW)
Create Vnf	Simple	Create a virtual network function (if a functional one is currently unavailable).	Mandatory: (vnf1) Optional: (endpoint1, endpoint2, vnf2, vnf3)
Apply Filter	Simple	Modify characteristics of a flow, for example, allow and block ports.	Mandatory: (action) Optional: (port, source, target)
Create Flow+Create Vnf	2-composite		Mandatory: (source, target, vnf1) Optional: (CPU, GPU, MEM, BW, endpoint1, endpoint2, vnf2, vnf3)
Create Flow+Apply Filter	2-composite		Mandatory: (source, target, action) Optional: (CPU, GPU, MEM, BW, port, source, target)
Create Flow+Create Vnf+Apply Filter	3-composite		Mandatory: (source, target, vnf1, action) Optional: (CPU, GPU, MEM, BW, endpoint1, endpoint2, vnf2, vnf3, port, source, target)

C. Transformer-based Intent Prediction

The Transformer-based Intent Prediction module is the cornerstone of our proactive approach. Its primary function is to anticipate future network demands and potential user intents before they explicitly arrive, thereby enabling the system to prepare and optimize resources in advance. This module continuously analyzes a rich stream of historical and real-time data, including: 1) Past intent requests and their characteristics; 2) Physical network status (e.g., available resources, topology).

The Transformer's self-attention mechanism is particularly well-suited for this task, as it can capture long-range dependencies and complex temporal relationships within the diverse input data. Unlike traditional time-series forecasting models that might focus solely on numerical metrics, the Transformer can also learn patterns related to the types of intents and their associated attributes that are likely to emerge. The output of this module is a set of "predicted intents," formatted similarly to the actual ingested intents, but with a future validity period. These predictions provide the RL-Agent with foresight into upcoming service requirements, allowing for proactive resource reservation, VNF pre-instantiation, or network path optimization. The accuracy of these predictions directly impacts the system's ability to minimize reactive reconfigurations and maintain high performance.

D. Intent-based SFC Placement

The Intent-based SFC Placement module, powered by a RL Agent, is responsible for making real-time, adaptive decisions on how to map the SFCs (derived from both current and predicted intents) onto the physical network infrastructure. This module operates within a Markov Decision Process (MDP) framework, where: 1) **State:** The state space encompasses the current "Physical Network Settings" (available CPU, GPU, memory, bandwidth on nodes and links) and the "Virtual Network Settings" (the characteristics of current and predicted SFC requests, including VNF types, their sequence, and QoS requirements); 2) **Action:** The RL-Agent's actions involve deciding which physical node to place a VNF instance on, and which physical path to map each virtual link between VNFs. In addition, the RL-Agent's actions also decide whether to accept or reject an SFC request if constraints cannot be met; 3) **Reward:** The reward function is designed to align with the overall objective function Equation III-B.

The RL-Agent learns an optimal policy through continuous interaction with the NFV Environment. It explores different placement strategies and receives feedback (rewards) based on the resulting network state and performance. This learning process allows the RL-Agent to adapt its decision-making in highly dynamic and uncertain IoT network conditions, effectively balancing immediate demands with future requirements predicted by the Transformer. The integration of predicted intents into the RL state space enables the agent to make proactive decisions, leading to more stable and efficient network operations compared to purely reactive approaches.

V. Experiments

Our experimental design encompasses both comparative experiments and ablation experiments. In the comparative experiments, we aim to observe the superiority of our proposed method over baseline comparison methods. The ablation experiments, on the other hand, are conducted to evaluate the significance of each component within our proposed method.

A. Experimental Setup

1) Datasets: We utilize the NLU Dataset [**?**] as shown in Table I. This dataset is a synthetically generated one and is commonly employed for training and developing natural language understanding models based on intent-driven networks. The dataset provides intent texts that describe common network operations. These intent texts are expressed in English and also encompass relevant network entities and attributes, such as IP addresses, the number of CPU cores, and network

TABLE II: Comparison Experiment

Method	Limitation	Metrics							
		BLEU↑	MDR↑	ROUGE-L↑	FEACI				
					Format↑	Explain↑	Accuracy↑	Normalized Cost↓	Normalized Inference↓
ChatGPT-4o	+ Prompt[ZERO]	0.413	0.653	0.465	0.00	0.00	0.00	**0.100**	**0.102**
	+ KOR	0.343	0.534	0.324	0.983	0.342	0.452	0.234	0.123
	+ Prompt[ONE]	**0.532**	0.783	0.437	0.755	0.643	0.642	0.334	0.331
	+ Prompt[FEW]	0.236	**0.833**	**0.593**	0.954	**0.957**	**0.843**	1.00	0.944
	+ GRAMMAR	0.433	0.682	0.432	**0.993**	0.354	0.331	0.342	0.945
Qwen3-32b	+ Prompt[ZERO]	0.455	0.774	0.342	0.00	0.00	0.00	0.00	**0.102**
	+ KOR	0.599	0.541	0.449	0.954	0.344	0.483	0.00	0.122
	+ Prompt[ONE]	0.313	0.853	0.593	0.86	0.775	0.874	0.00	0.323
	+ Prompt[FEW]	**0.656**	**0.943**	**0.599**	0.955	**0.974**	**0.934**	0.00	0.983
	+ GRAMMAR	0.549	0.674	0.483	**0.996**	0.314	0.328	0.00	0.954
Deepseek-v3	+ Prompt[ZERO]	0.482	0.768	0.389	0.00	0.00	0.00	0.00	**0.105**
	+ KOR	0.624	0.753	0.490	0.945	0.323	0.442	0.00	0.231
	+ Prompt[ONE]	0.231	0.764	0.483	0.809	0.654	0.722	0.00	0.332
	+ Prompt[FEW]	**0.665**	**0.892**	**0.594**	0.964	**0.932**	**0.889**	0.00	1.00
	+ GRAMMAR	0.568	0.762	0.435	**0.992**	0.284	0.443	0.00	0.945

bandwidth. For example, "I need a virtual network with 5 virtual nodes and 8 virtual links, requiring 10 units of CPU at node A and 20 Mbps bandwidth on link B, ensuring low latency."

TABLE III: Transformer Model Configuration

Parameter	**Value**
Padding token index (source)	1
Padding token index (target)	1
Start-of-sequence token index (target)	2
Encoder vocabulary size	431
Decoder vocabulary size	431
Number of attention heads	8
Number of encoder/decoder layers	6
Dropout probability (hidden layers)	0.1
Total trainable parameters	44,800,943

2) Hyperparameters: We use Proximal Policy Optimization (PPO) [28] for resource allocation, which is a popular and empirically effective actor-critic algorithm known for its stability and data efficiency, achieved by clipping the objective function.

3) Transformer model: In this experiment, the Transformer model we adopted is configured as shown in Table III. The padding token indices for both the source and target sequences are set to 1, while the start-of-sequence token index for the target sequence is 2. Both the encoder and decoder have a vocabulary size of 431. The model employs 8 attention heads and consists of 6 layers for both the encoder and decoder. A dropout probability of 0.1 is applied in the hidden layers to prevent overfitting. The model contains a total of 44,800,943 trainable parameters. By leveraging multi-head attention mechanisms and feedforward networks, this model achieves efficient sequence-to-sequence task modeling.

4) Environment: In our experiment, we constructed a physical network environment. The specific experimental conditions are as follows: For the network topology structure, we set the number of nodes to 100 to simulate a medium-sized physical network. The topology type selected was the Waxman model, which can effectively simulate the non-uniform distribution of nodes and the irregularity of connections in real-world networks. For the computing resources of nodes, we defined the values of this attribute are generated using a uniform distribution. The value range is between 50 and 100. For the bandwidth resources of links, we defined the values of this attribute are also generated using a uniform distribution, with a data type of integer, set to be generative, and a value range between 50 and 100.

5) Metrics: We use following metrics in our experiments:

(1) Long-term Revenue-to-Cost Ratio (LRC). The LRC evaluates the economic efficiency of the NFV-RA strategy by comparing the cumulative revenue generated from accepted VNs to the cumulative cost of the resources consumed for their embedding over a period. A higher LRC signifies greater profitability and resource efficiency. It is formulated as:

$$LRC = \frac{\Sigma_{t=0}^{\tau}\Sigma_{I\in\hat{I}(t)}REV(S)\times\overline{w}}{\Sigma_{t=0}^{\tau}\Sigma_{I\in\hat{I}(t)}COST(S)\times\overline{w}} \tag{6}$$

where, S is the embedding solution for an instance I, $REV(S)$ is the revenue generated by embedding VN g_v, $COST(S)$ is the resource consumption, $\overline{w}$ is the lifetime of the corresponding VN.

(2) Average Solving Time (AST). The Average Solving Time (AST) measures the average computational time (typically in seconds) an NFV-RA algorithm takes to find a solution for a single VN request or simulation run. This metric is crucial for assessing the algorithm's efficiency and its suitability for online, real-time decision-making environments. Lower AST values are generally preferred, especially for dynamic scenarios.

(3) Bilingual Evaluation Understudy (BLEU) score. BLEU score is a commonly used automated evaluation metric in the field of natural language processing, designed to measure the similarity between generated text and reference text.

$$BLEU = BP\cdot\exp\left(\sum_{n=1}^{N}w_n\log P_n\right) \tag{7}$$

where BP is the brevity penalty factor. P_n represents the n-gram precision between the generated text and the reference

text. w_n is the weight, which is typically assigned equally to different n-grams.

(4) Manunal Double-check Rate (MDR). As LLMs may generate multiple answers, we also conducted a manual double-check finally , to judge if wrong intent are included [29]. We calculate the accuracy by determining the proportion of test intents that are correctly processed by the LLM among a number of test intents, and use this accuracy value as the value of the MDR.

(5) ROUGE-L: ROUGE-L is an NLP metric for evaluating text generation quality, measuring similarity via the Longest Common Subsequence (LCS). It focuses on content overlap and fluency, balancing precision and recall through F1 scoring. Unlike n-gram-based methods, it tolerates word order variations, making it robust for summarization and translation tasks.

(6) FEACI [30]: FEACI is a novel evaluation metric designed to assess LLM-generated responses in intent translation/resolution tasks, addressing limitations of traditional metrics like BLEU and ROUGE. It evaluates five key dimensions: Format (structural correctness), Explanation (quality of reasoning), Accuracy (value matching with references), Cost (token-based pricing for closed-source models), and Inference Time (response generation delay). Each dimension is scored (0-1) and combined via weighted summation, where weights reflect their relative importance. This holistic approach ensures balanced evaluation of semantic understanding, practicality, and efficiency, making it suitable for telecom and other domain-specific applications.

B. Comparison Experiment

To ensure that the LLM outputs the requirements for configuring virtual networks in a fixed .Yaml format after interpreting user intents, we employed four different methods to restrict the model's output:

- **Prompt[ZERO]:** Without providing any additional contextual information to the LLMs, we evaluate their ability to generate technical intents in "standard" formats (e.g., 3GPP and TMF specifications) under a zero-shot (ZERO-shot) setting, where such formats may not have been encountered during the model's training phase.
- **Prompt[ONE]:** In Oneshot prompting scenarioswe provide additional examples including only the expected response to the LLM models, explicitly specifying the expected results in YAML format.
- **Prompt[FEW]:** In Few-shot prompting scenarios, we provide LLM models with additional examples, explicitly specifying the expected results in YAML format. Moreover, these examples further illustrate how to calculate specific fields in the expected response based on the technical intents outlined in the service order.
- **GRAMMAR:** Utilize grammatical rules to compel the model's output.
- **KOR:** Extract structured data from the text generated by the LLM.

Moreover, we conducted tests using three different LLM models:

- **ChatGPT-4o:** a cutting-edge model with enhanced context understanding and multi-modal support.
- **Qwen3-32b:** a cutting-edge model with enhanced context understanding and multi-modal support.
- **Deepseek-v3:** an advanced model optimizing response accuracy and knowledge retrieval.

We send requests to these models via API interfaces, and they typically possess more trainable parameters compared to the open-source models locally deployed on our computing server. We demonstrate the inference cost differences among various large language models by comparing the expenses (in USD) required to process 1 million input/output tokens.

In terms of the decoding strategy for language models to generate network configurations, we employ temperature sampling and nucleus sampling methods, with the temperature parameter set to 0.2 and the top-p threshold set to 0.9. Table IV lists the specific parameters of the LLMs used in this study.

TABLE IV: Comparison of Model Parameters

Parameter(s)	GPT4-o	Qwen3-32b	Deepseek-v3
N-params	1760B	32B	67B
Open	N	Y	Y
Attention heads	32	32	64
Attention layer(s)	128	32	80
Activation function	SwiGLU	SwiGLU	SwiGLU
Cost In/Out (per 1M TK)	$10/$30	$0/$0	$0/$0
Training Data Scale	13T tokens	3T tokens	8T tokens
Context Window	128K	32K	128K
Inference Speed (tokens/s)	120	350	200

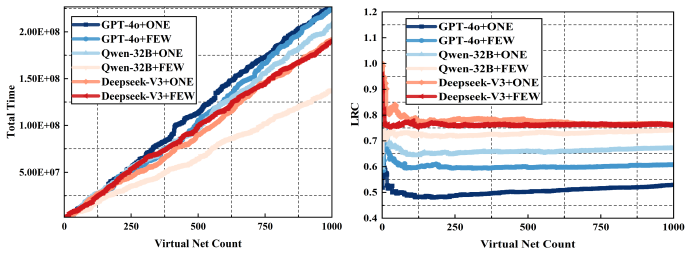

Fig. 2: Comparison of total time and LRC between different LLM models under the same user intent

Results Analysis: From the perspective of different restrictive condition methods, for the three LLM models, the Format and Accuracy metrics perform better under the Prompt[FEW] condition. Under the Prompt[ZERO] condition, they only perform well in terms of Normalized Cost and Normalized Inference Time, meaning lower costs and shorter inference times. However, the model's accuracy and capture of user intent are somewhat lacking. As for the GRAMMAR and KOR restrictive methods, although LLMs can achieve the highest values in the Format metric, their output content lacks interpretability and cannot be used for intent data modeling.

From the perspective of different model methods, in terms of overall performance, Qwen-32B and Deepseek-v3 perform relatively closely. Both can reach high values in Explain and Accuracy, and outperform ChatGPT-4o in terms of Format and Accuracy. Nevertheless, ChatGPT-4o and Deepseek-v3 have lower costs and shorter inference times under the

Prompt[ZERO] condition, while Qwen-32B shows relatively stable performance in terms of cost and inference time under different conditions.

Therefore, it can be summarized that adopting the Prompt[FEW] condition generally helps to improve various performance metrics of the models, and using Deepseek-v3 offers the most stable performance.

C. Ablation Study

This task can be formulated as a text generation task. In the experiment, we first execute the algorithm for 200 rounds to collect sufficient data for predicting the user's intent. We use the BLEU score to evaluate the similarity between the intent predicted by the Transformer and the real intent.

We proposed three different baselines:

- **NLU:** Utilizes only features from the user's historical intent data.
- **NLU+V_Net:** Utilizes features from the user's historical intent data and fuses features generated by an extra graph convolutional network on Visual net configuration.
- **NLU+P_Net:** Utilizes features from the user's historical intent data and fuses features generated by an extra graph convolutional network Physical net configuration.

Results Analysis: As shown in Fig. 3, the NLU+V_Net method converges rapidly and achieves the highest BLEU score, indicating successful prediction of user intent based on historical intents and virtual network configuration. In contrast, the NLU+P_Net method showed the lowest BLEU score and exhibited instability during training. We posit that fusing physical network configuration features does not strengthen the representation of the user's intent features.

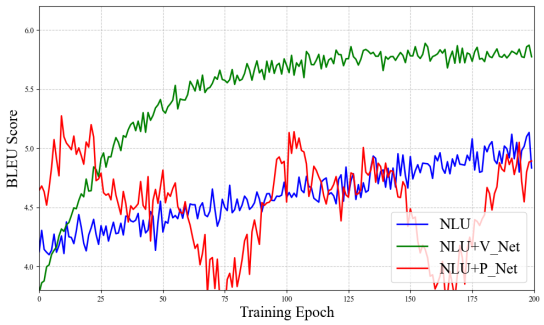

Fig. 3: Intent Prediction Performance Comparison.

VI. Conclusion and Future Work

This paper has introduced a novel proactive intent-driven SFC placement framework specifically designed for the challenges of dynamic IoT networks. By integrating Transformer-based intent prediction, LLM-driven intent translation, and Reinforcement Learning for dynamic SFC deployment, our approach moves beyond the limitations of current reactive methods. The proposed framework offers significant advantages by enabling the network to anticipate future demands, interpret complex human intents with greater flexibility, and make real-time, adaptive placement decisions. This synergy leads to enhanced resource utilization, improved SFC acceptance rates, and a more resilient network infrastructure capable of meeting stringent QoS requirements in highly volatile IoT environments. Our integrated methodology promises a more autonomous and efficient paradigm for managing network services.

For future work, we plan to extend our research in several directions. Firstly, we aim to explore the development of more sophisticated Transformer architectures for intent prediction, potentially incorporating multi-modal data (e.g., environmental sensor readings, social media trends) to further enhance prediction accuracy and lead time. Secondly, we will investigate methods for real-time fine-tuning and continuous learning for the LLM, allowing it to adapt to evolving intent expressions and new service requirements without extensive retraining. Thirdly, we intend to explore multi-agent RL approaches to handle distributed SFC placement decisions across a large-scale, hierarchical IoT network, considering inter-SFC dependencies and potential conflicts. Finally, we plan to conduct extensive evaluations in a real-world IoT testbed to validate the framework's performance under authentic traffic conditions and assess its scalability and robustness in practical deployments.

References

[1] M. Noaman, M. S. Khan, M. F. Abrar, S. Ali, A. Alvi, and M. A. Saleem, "Challenges in integration of heterogeneous internet of things," *Scientific Programming*, vol. 2022, no. 1, p. 8626882, 2022.

[2] Z. Huang and Y. Wu, "A survey on explainable anomaly detection for industrial internet of things," in *2022 IEEE Conference on Dependable and Secure Computing (DSC)*, pp. 1–9, 2022.

[3] E. F. Maleki, W. Ma, L. Mashayekhy, and H. J. La Roche, "Qos-aware content delivery in 5g-enabled edge computing: Learning-based approaches," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9324–9336, 2024.

[4] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "Unlocking efficiency in b5g networks: The need for adaptive service function chains," in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–8, IEEE, 2024.

[5] X. Zhou, X. Ye, K. I.-K. Wang, W. Liang, N. K. C. Nair, S. Shimizu, Z. Yan, and Q. Jin, "Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1742–1751, 2023.

[6] L. Velasco, M. Signorelli, O. G. De Dios, C. Papagianni, R. Bifulco, J. J. V. Olmos, S. Pryor, G. Carrozzo, J. Schulz-Zander, M. Bennis, *et al.*, "End-to-end intent-based networking," *IEEE communications Magazine*, vol. 59, no. 10, pp. 106–112, 2021.

[7] J. Andrade-Hoz, Q. Wang, and J. M. Alcaraz-Calero, "Infrastructure-wide and intent-based networking dataset for 5g-and-beyond ai-driven autonomous networks," *Sensors*, vol. 24, no. 3, p. 783, 2024.

[8] M. Avgeris, A. Leivadeas, N. Athanasopoulos, I. Lambadaris, and M. Falkner, "Model predictive control for automated network assurance in intent-based networking enabled service function chains," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–7, IEEE, 2023.

[9] A. Leivadeas and M. Falkner, "Vnf placement problem: A multi-tenant intent-based networking approach," in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 143–150, 2021.

[10] M. Avgeris, A. Leivadeas, N. Athanasopoulos, I. Lambadaris, and M. Falkner, "Model predictive control for automated network assurance in intent-based networking enabled service function chains," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–7, 2023.

[11] H. Hantouti, N. Benamar, and T. Taleb, "Service function chaining in 5g & beyond networks: Challenges and open research issues," *IEEE Network*, vol. 34, no. 4, pp. 320–327, 2020.

[12] J. Liu, G. Shou, Q. Wang, Y. Liu, Y. Hu, and Z. Guo, "Load-balanced service function chaining in edge computing over fiwi access networks for internet of things," *arXiv preprint arXiv:2006.08134*, 2020.

[13] Z. Shi, Y. Zeng, and Z. Wu, "Service chain orchestration based on deep reinforcement learning in intent-based iot," in *Proceedings of the 9th International Conference on Computer Engineering and Networks*, pp. 875–882, Springer, 2020.

[14] M. Chowdhury, "Accelerator: an intent-based intelligent resource-slicing scheme for sfc-based 6g application execution over sdn-and nfv-empowered zero-touch network," *Frontiers in Communications and Networks*, vol. 5, p. 1385656, 2024.

[15] S. Zou, M. Liwang, B. Wu, W. Wu, Y. Sun, and W. Ni, "Intent-oriented network slicing with hypergraphs," *IEEE Network*, 2024.

[16] S. Moazzeni, Z. Huang, S. Zeb, X. Zhang, J. P. Ullauri, A. Bravalheri, R. Hussain, Y. Wu, X. Vasilakos, and D. Simeonidou, "Federated intelligent service function chain orchestration in future 6g networks," *Authorea Preprints*, 2025.

[17] T. Wang, Q. Fan, C. Wang, L. Ding, N. J. Yuan, and H. Xiong, "Flagvne: A flexible and generalizable reinforcement learning framework for network resource allocation," in *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, 2024.

[18] J. Wang, J. Hu, G. Min, W. Zhan, A. Y. Zomaya, and N. Georgalas, "Dependent task offloading for edge computing based on deep reinforcement learning," *IEEE Transactions on Computers*, vol. 71, no. 10, pp. 2449–2461, 2021.

[19] X. Zhang, Y. Ma, Z. Huang, Y. Wu, R. Hussain, S. Moazzeni, and D. Simeonidou, "Cross-domain low latency e2e network service delivery with federated and transfer learning," *Authorea Preprints*, 2025.

[20] J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 242–253, 2020.

[21] X. Zhang, J. M. Parra-Ullauri, S. Moazzeni, X. Vasilakos, R. Nejabati, and D. Simeonidou, "Federated analytics with data augmentation in domain generalization towards future networks," *IEEE Transactions on Machine Learning in Communications and Networking*, 2024.

[22] X. Zhou, W. Huang, W. Liang, Z. Yan, J. Ma, Y. Pan, and K. I.-K. Wang, "Federated distillation and blockchain empowered secure knowledge sharing for internet of medical things," *Information Sciences*, vol. 662, p. 120217, 2024.

[23] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630–641, 2021.

[24] J. Wang, J. Hu, J. Mills, G. Min, M. Xia, and N. Georgalas, "Federated ensemble model-based reinforcement learning in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 6, pp. 1848–1859, 2023.

[25] N. Van Tu, J.-H. Yoo, and J. W.-K. Hong, "Towards intent-based configuration for network function virtualization using in-context learning in large language models," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pp. 1–8, IEEE, 2024.

[26] S. Alam and W.-C. Song, "Enhancing network intelligence with llm-based ibn and drl: A dynamic approach for sagin resource management," in *2025 International Conference on Computing, Networking and Communications (ICNC)*, pp. 723–727, IEEE, 2025.

[27] A. Mekrache and A. Ksentini, "Llm-enabled intent-driven service configuration for next generation networks," in *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, pp. 253–257, IEEE, 2024.

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

[29] Y. Tan, D. Min, Y. Li, W. Li, N. Hu, Y. Chen, and G. Qi, "Evaluation of chatgpt as a question answering system for answering complex questions," *arXiv preprint arXiv:2303.07992*, 2023.

[30] L. Dinh, S. Cherrared, X. Huang, and F. Guillemin, "Towards end-to-end network intent management with large language models," *arXiv preprint arXiv:2504.13589*, 2025.