electronics

MDPI

Article

When Large Language Models Meet Optical Networks: Paving the Way for Automation

Danshi Wang *, Yidi Wang, Xiaotian Jiang, Yao Zhang, Yue Pang and Min Zhang

State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China; wangyidi@bupt.edu.cn (Y.W.); jxt@bupt.edu.cn (X.J.); zhang-yao@bupt.edu.cn (Y.Z.); pangyue@bupt.edu.cn (Y.P.); mzhang@bupt.edu.cn (M.Z.)

* Correspondence: danshi_wang@bupt.edu.cn

Abstract: Since the advent of GPT, large language models (LLMs) have brought about revolutionary advancements in all walks of life. As a superior natural language processing (NLP) technology, LLMs have consistently achieved state-of-the-art performance in numerous areas. However, LLMs are considered to be general-purpose models for NLP tasks, which may encounter challenges when applied to complex tasks in specialized fields such as optical networks. In this study, we propose a framework of LLM-empowered optical networks, facilitating intelligent control of the physical layer and efficient interaction with the application layer through an LLM-driven agent (AI-Agent) deployed in the control layer. The AI-Agent can leverage external tools and extract domain knowledge from a comprehensive resource library specifically established for optical networks. This is achieved through user input and well-crafted prompts, enabling the generation of control instructions and result representations for autonomous operation and maintenance in optical networks. To improve LLM's capability in professional fields and stimulate its potential on complex tasks, the details of performing prompt engineering, establishing domain knowledge library, and implementing complex tasks are illustrated in this study. Moreover, the proposed framework is verified on two typical tasks: network alarm analysis and network performance optimization. The good response accuracies and semantic similarities of 2400 test situations exhibit the great potential of LLM in optical networks.

Keywords: large language model; optical network automation; prompt engineering

check for updates

Citation: Wang, D.; Wang, Y.; Jiang, X.; Zhang, Y.; Pang, Y.; Zhang, M. When Large Language Models Meet Optical Networks: Paving the Way for Automation. *Electronics* **2024**, *13*, 2529. https://doi.org/10.3390/electronics13132529

Academic Editor: Martin Reisslein

Received: 16 May 2024
Revised: 24 June 2024
Accepted: 25 June 2024
Published: 27 June 2024

1. Introduction

A large language model (LLM) is a type of natural language processing (NLP) technology that relies on deep learning with an extensive number of parameters and acquire abundant knowledge from large-scale datasets [1]. The emergence of language models can be traced back to the 1950s and 1960s, when the foundations of NLP were initially established. Early endeavors in this field were propelled by rule-based systems and statistical methodologies. Before the advent of deep learning, Hidden Markov Models and N-gram models stood out as two popular approaches for NLP. In recent years, LLMs have achieved remarkable progress with the introduction of the Transformer architecture in 2017 [2].

Since then, numerous Transformer-based LLMs have been developed by several giant AI companies, such as Google's T5 and Gemini, Meta's LLaMa, Hugging Face's BLOOM, and notably OpenAI's Generative Pre-training Transformer (GPT). As a generative AI, GPT is currently considered as the most prominent and influential LLM, which can provide quick and reasonable answers to questions with human-like conversations and have achieved two major milestones: ChatGPT and GPT-4 [3]. GPT has achieved state-of-the-art results on many NLP benchmarks and applications, such as intent understanding, content generating, text summarization, language translating, question answering, code debugging, and conversational robots in the fields of healthcare, education, law, finance, customer service, search engines, and scientific research. Typically, GPT is a generalist model that can effectively solve multiple general-purpose NLP tasks on a broad scale. However, it still faces challenges when tackling specific or complex tasks within particular professional domains [4]. In such cases, it is imperative to devise suitable task instructions or formulate specific prompt strategies to elicit its specialized proficiencies.

As the cornerstone of informative infrastructures, optical networks are constructed with cross-connected optical fibers for massive data transmission between multiple nodes, with the characteristics of high capacity, long distance, large scale, and widespread deployment. Optical networks comprise lots of physical components, including optical transceivers, fiber optic cables, optical amplifiers, optical filters, optical switches, and other components. The categories of optical network can be classified as access network, metro network, and core network in terms of the service scope. In view of the fact that massive components, various topologies, and multiple scales, the operation and maintenance of optical networks are intricate and burdensome, including network monitoring, controlling, optimizing, upgrading, scheduling, troubleshooting, etc. [5]. Current network operation and maintenance rely heavily on human labors, involving a substantial workforce (domain specialist, network engineer, network administrator, network analyst, site worker, service personnel, and other staff) engaged in specific and tedious tasks. Therefore, it is highly anticipated to develop an intelligent operation agent to assist network operators and engineers in diverse tasks [6], ultimately leading to the achievement of autonomous optical networks [7].

The past few years have witnessed a remarkable surge in research efforts from both academia and industry to seamlessly integrate and harness the potential of AI across various aspects of optical networks [8]. However, previous studies have mainly relied on conventional machine learning or deep learning models with relatively small sizes. These small-sized models exhibit limited intelligence, and can only perform a few specific functions, but falling short of achieving artificial general intelligence (AGI) and full automation capabilities. Meanwhile, extensive research on large models has shown that when scaling up the model size (over tens of billions of parameters) with augmented training data, the model capability can be significantly enhanced on a wide range of downstream tasks [9]. The advancement of LLM is paving the way for the realization of AGI and prompts us to expect the prospects of autonomous optical networks.

In this paper, encouraged by the leapfrog development and strong capabilities of LLMs, a framework of LLM-empowered optical networks is proposed. As the core component, an LLM-driven agent (AI-Agent) is deployed in the control layer, which can generate appropriate responses based on the user's input, well-crafted prompts, and domain resource library. Through the generated control instructions and result representation, the AI-Agent facilitates autonomous control of the physical layer and efficient interaction with the application layer. To enhance LLM's specific capability in optical networks and maximize its potential on complex tasks, we have devised prompt engineering strategies and established a comprehensive domain resource library for the implementation of such tasks. Moreover, two typical tasks in optical networks (alarm analysis and autonomous optimization) are demonstrated as the case study of the proposed framework. The test results show that the application of LLM in optical networks has the great potential to simplify the operational processes and improve management automation. Furthermore, our work illustrates that LLMs can pave the way to a more intelligent, efficient and autonomous future in various fields, including optical networks.

The rest of the paper is organized as follows. In Section 2, we provide a detailed introduction to the framework of LLM-empowered optical networks, which enables intelligent control of the physical layer and efficient interaction with the application layer. In Section 3, the prompt engineering consisting of prompt elements and prompt techniques is introduced in detail. In Section 4, we describe the composition of a domain resource library for optical networks and how to invoke tools and extract information from it. In Section 5, a five-step procedural framework for solving the complex tasks in optical networks is demonstrated. In Section 6, two case studies of alarm analysis and autonomous optimization are studied to illustrate the feasibility and effectiveness of LLM's applications in optical networks. Finally, conclusions for this paper are provided in Section 7.

2. Framework of LLM-Empowered Optical Networks

The proposed framework of LLM-empowered optical networks is illustrated in Figure 1. The hierarchy of optical networks can be divided into access, metro, core, and data center (DC) networks. Moreover, optical networks are progressively advancing towards large-scale deployment, massive fiber-optic infrastructures, dynamic reconfiguration capabilities, ultra-wideband transmission, software-defined control mechanisms, and resource virtualization, as well as open and disaggregated architectures. These emerging trends pose the challenges to the operation and maintenance of optical networks, including network planning, link optimization, hardware configuration, software development, failure management, system upgrades, etc.

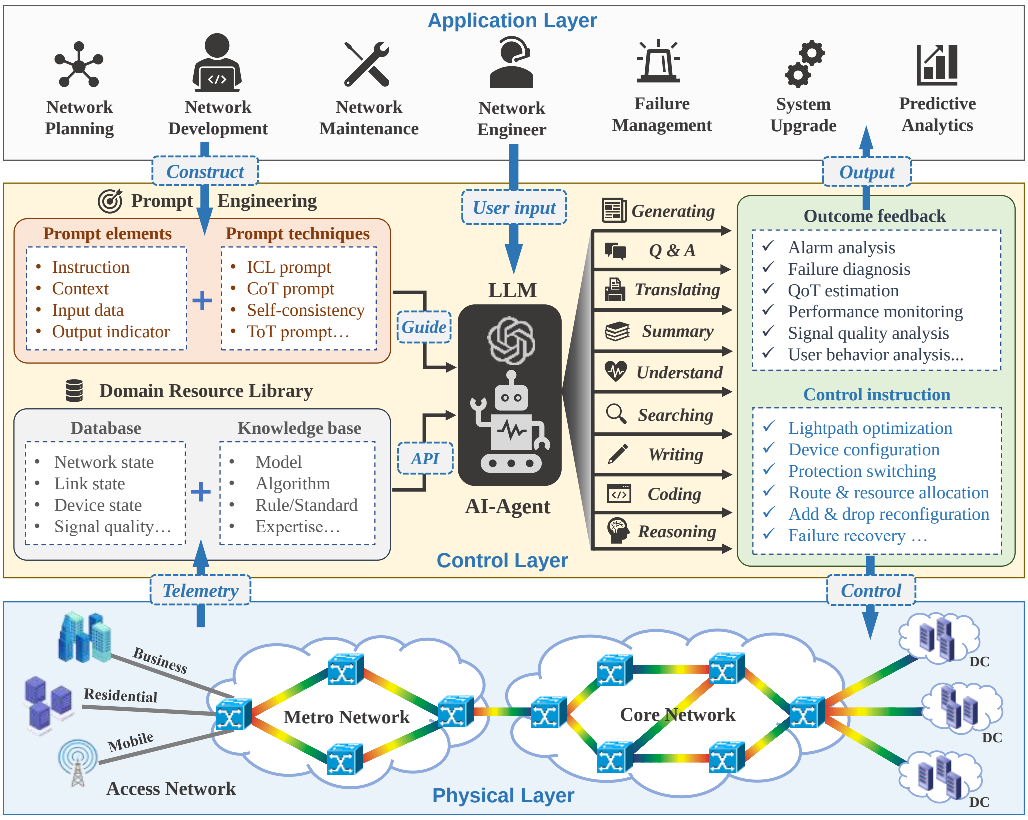


Figure 1. The framework of LLM-empowered optical networks, which enables intelligent control of the physical layer and efficient interaction with the application layer through an LLM-driven AI-Agent deployed in the control layer.

To address these intractable issues, an LLM-driven intelligent assistant, termed "AI-Agent", is designed and deployed in the control layer, as shown in Figure 1. Its purpose is to assist network operators in facilitating intelligent control on the physical layer and efficient interaction with the application layer. In practical applications of such intelligent control, the AI-Agent is placed along with the software-defined network controller in the network layer. The LLM-generated commands are forwarded to physical layer devices through the Southbound Interface using standards such as the Network Configuration Protocol. In this paper, the GPT-series models are selected as the representative model for a case study. Different from conventional deep learning models that necessitate retraining for each task, GPT is a pre-trained LLM that has learned from extensive data that can directly perform various general-purpose tasks without the need for additional training. However, it always encounters challenges when applied to specific tasks in professional domains, and may fail to deal with complex issues in optical networks. In order to enhance professional capabilities of LLM for optical networks, it is imperative to augment LLM through the implementation of "*Prompt Engineering*" and "*Retrieval-augmented Generation (RAG) [10]*".

A "Prompt" is a sequence of tokens that serves as textual inputs to an LLM, instructing it to enhance its capability for a specific task. Prompt engineering is an emerging discipline for developing and optimizing prompt to elicit the desired output from an LLM [11]. It should be noted that for non-open source LLMs (e.g., ChatGPT and GPT-4), we have little to do in the training or retraining phases. In such cases, prompt engineering emerges as the most prominent step that we can perform in developing LLMs for specialized applications. In optical networks, by crafting effective prompts, we can leverage the power and potential of LLMs for various purposes, such as answering operator's queries for troubleshooting, understanding user's intent, translating natural language into machine commands, generating control instructions for device configuration, summarizing useful information from massive network data, searching solutions from resource library, recording and reading network log, writing or debugging codes, reasoning failure cause, developing optimization strategy, etc.

The RAG is a technique for enhancing the accuracy and reliability of LLMs with facts fetched from external sources, which references an authoritative domain resource library outside of its training data sources before generating a response [12]. The domain resource library consists primarily of two parts: database and knowledge base. In optical networks, significant amounts of data are generated from a wide range of sources, including optical performance monitoring data, network condition data, device state data, alarm data, traffic data, and service data. These raw data can be collected in real-time through telemetry to control plane. All sorts of collected data comprehensively represent the network operating states and can be leveraged by the LLM-driven AI-Agent for information acquisition and state awareness. In addition, the field of optical networks have accumulated a wealth of knowledge base, encompassing diverse models, mathematical theories, physical mechanisms, operation rules, standard protocols, expertise, and other instructive information. These tools in knowledge base can be readily accessed by the AI-Agent through the application programming interface (API) to execute customized functions for the assigned tasks or provide expert knowledge for analysis and decision.

As described above, constructing effective prompts and leveraging appropriate domain resource can activate diverse potential capabilities of LLM. In optical networks, the LLM-driven AI-Agent is envisioned to play a pivotal role in autonomous network control and intuitive outcome feedback, facilitating efficient interaction across physical layer, control layer, and application layer, as depicted in Figure 1. When implementing the AI-Agent, the network operators can directly input their intents, tasks, or questions to the AI-Agent in the manner of conversation. According to user's input, the AI-Agent will generate the corresponding response in the form of text, enabling the desired action through API integration and external tools. The control instruction that interacts with physical layer can be generated by the AI-Agent to automatically execute tasks, such as lightpath optimization, device configuration, protection switching, failure recovery, route and resource allocation, impairment mitigation, etc. The outcome feedback that interact with application layer are output from the AI-Agent to intuitively present the processed information and analysis results, such as alarm analysis, failure diagnosis, quality of transmission (QoT) estimation, performance monitoring, signal quality analysis, traffic prediction, and user behavior analysis. Therefore, in optical networks, the LLMs hold significant potential to accomplish various tasks for autonomous operation and active maintenance, while also facilitating a paradigm shift in human-network interaction.

3. Prompt Engineering for LLM

Prompt engineering is an art aimed at crafting effective prompts that guide LLMs to generate desired responses. As a fundamental technique in NLP, it plays a crucial role in maximizing the effectiveness of LLMs by bridging the gap between user intent and model comprehension [11]. To harness the full potential of LLMs, a well-designed prompt is indispensable, which requires a good understanding and mastery of the basic elements and techniques of prompt engineering.

3.1. Prompt Elements

To effectively convey the user's intent to the LLM, a comprehensive and coherent prompt is generally required, consisting of four basic elements: instruction, context, input data, and output indicator. As the central element of prompt, instruction helps LLM construct logical reasoning chains by providing clear guidance on what the user wants, directly connecting the user intent to model comprehension. For further inference and analysis, necessary context would be helpful, which can be external information or additional background that guide the LLM to respond better. In more complex applications, instruction-related data are also part of the prompt, which drives the LLM to process the specific scenario user interests. Additionally, the output indicator is required when the LLM's output is desired to be of a specific type or format. It should be noted that a well-designed prompt does not have to contain all four elements. Depending on the complexity and requirements of the application scenario, the prompt may contain three, two, or even just the instruction element.

3.2. Prompt Techniques

In addition to prompt elements, another key point of prompt engineering is the prompt techniques, which play a pivotal role in maximizing the information of prompt elements, and stimulating the LLM's analyzing and reasoning potential to obtain high-quality output. The schematics of some typical prompting techniques are shown in Figure 2. As the fundamental prompt technique, zero-shot in-context learning (ICL) [13] is a simple and straightforward way to convey user's intent to LLM by providing only instructions as input. However, it usually performs poorly in complex applications, since the pre-trained knowledge can hardly satisfy the user-defined scenario. To address this problem, few-shot ICL emerges as a recommended solution, which yields promising results in intricate tasks by guiding the LLM learn from similar examples from domain resource library and input data. However, there still exist limitations in complex tasks using few-shot ICL, where multi-step reasoning and analyzing are required. By decomposing complex reasoning tasks into multiple simple subtasks, the chain of thought (CoT) [14] prompting enables the LLM to think and reason step by step, thereby overcoming the limitation of insufficient reasoning ability. However, diverse reasoning paths may lead to different reasoning results. To ensure the output aligns with the user's intent, self-consistency is usually incorporated into CoT, which selects the most consistent answer as the final output according to the majority vote on different reasoning paths. Furthermore, the evaluation ability and search algorithm can also be introduced in each step of the chain, and the best step candidates are reserved for the next step of reasoning. Such tree of thoughts (ToT) prompting allows LLM to self-evaluate the progress that intermediate steps made towards completing the task through a deliberate reasoning process. Moreover, emerging prompting techniques such as automatic prompting, active prompting, and directional stimulus prompting have also been proposed recently for different types of tasks.

The successful implementation of LLM for desired tasks in optical networks heavily relies on the meticulous selection of prompt elements and seamless integration of prompt techniques, signifying the paramount significance of prompt engineering. Prompt engineering is an empirical science and the effect of prompt engineering methods can vary a lot among models, thus requiring heavy experimentation and heuristics. Consequently, it can be inferred that the future research on LLM will primarily strive to explore advanced techniques in prompt engineering.

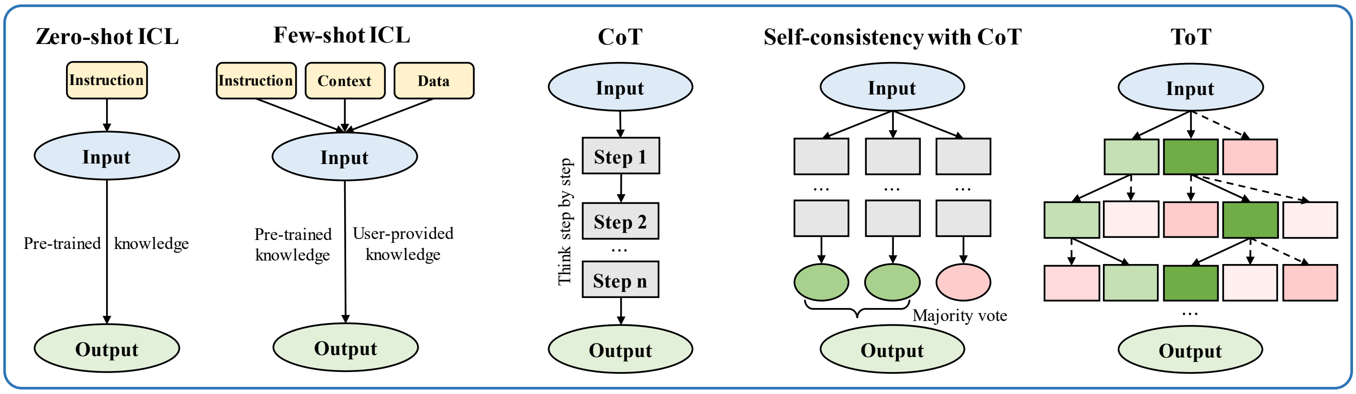


Figure 2. Schematic of some typical prompting techniques, including zero-shot ICL, few-shot ICL, CoT, self-consistency with CoT, and ToT.

4. Domain Resource Library

In addition to prompt engineering, it is also significant to take full advantage of specialized domain resources for specific tasks in optical networks. To provide accurate and professional assistance in optical networks, it is essential to establish and maintain a comprehensive domain resource library. This library should be easily accessible and readily available for LLMs, consisting of both a database and a knowledge base, as depicted in Figure 3. By extracting useful information from the database, it becomes convenient to search and provide tailored responses to specific problems, effectively activating its capability for targeted queries and tasks in optical networks; by invoking specialized model or expertise from the knowledge base, it can seamlessly integrate with LLM to create a more robust and versatile tool for targeted support and elaborate analysis.

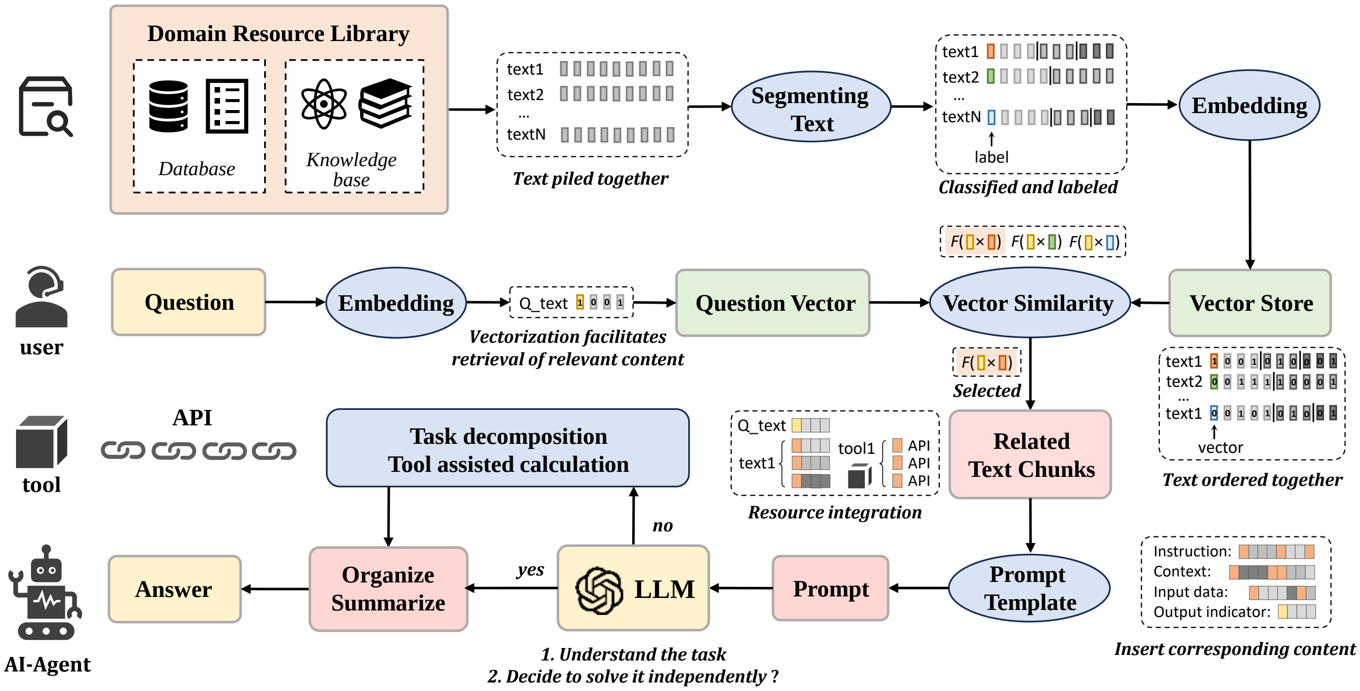


Figure 3. Workflow of LLM to invoke tools and extract information from the domain resource library for executing specific tasks in optical networks.

4.1. Data Base

In optical networks, significant amounts of data are generated from a wide range of sources, including historical data and real-time updated data that characterize the performance and states of equipment, link, and network. The equipment state data comprises measurements, such as gain, tilt, and the noise figure from an Erbium-doped fiber amplifier (EDFA); center wavelength, bandwidth, and attenuation values for each channel from the wavelength selective switch (WSS); and bit error rate (BER) and launch and received power from the optical module. The link performance data contains a fiber loss profile obtained through the optical time domain reflectometer (OTDR), and the optical power of each channel measured by the optical channel monitor (OCM). The network condition data encompasses information regarding the topology, traffic, resource utilization, routing, alarms, and other relevant aspects. These data can be collected by telemetry through optical supervisor channel (OSC) and conveniently viewed through a network management system.

4.2. Knowledge Base

In recent decades, the field of optical networks has accumulated a wealth of professional knowledge, providing effective external tools for LLMs. This domain knowledge base contains diverse models, algorithms, mechanisms, rules, protocols, and expertise [15]. In the physical layer, the typical and widely-used tools include Gaussian noise (GN) model for QoT estimation, split-step Fourier method (SSFM) for fiber channel modeling, digital signal processing (DSP) algorithms in coherent detection for impairment compensation, EDFA and WSS configuration algorithms for system optimization. In the network layer, the effective and helpful tools include routing and wavelength assignment (RWA) algorithms, resource allocations strategies, traffic prediction algorithms, alarm analysis rules, fault management techniques, system upgrade operations, and network maintain guidance. All of these tools can be accessed through an API between the knowledge base and LLM.

4.3. Invoking Tools and Extracting Information from the Domain Resource Library

The domain resource library encompasses a variety of content types, including words, tables, numerical values, formulas, charts, documents, and codes, all of which can be represented in the form of text. Therefore, when establishing a local domain resource library, additional methods are required to systematically organize the above types of content, as shown in Figure 3. First, the raw text data should be segmented into text chunks and ensure that each chunk is within the token limit. Then, the vectors of segmented text chunks are extracted into a vector store by embedding method constructed by LangChain [16] in parallel, aiming to facilitate finding the most relevant information as prompts of the potential to activate LLM.

The user's query is transformed into a question vector for the purpose of indexing. By calculating the similarity between the question vector and the existing content in the vector store, we can retrieve and select one or more closely related text chunks based on their similarity. The extracted text chunks may include data, knowledge, algorithms, and models. These contents will be organized and presented in a sequential manner to create prompts for activating the specific capabilities of the LLM, thereby generating corresponding responses. Inserting relevant content into the prompt template can help the LLM better understand tasks and determine whether they can be directly solved with no need of task decomposition or external resources. According to the LLM's response, the tasks will be categorized into two groups: simple tasks that can be solved directly, and complex tasks that require further decomposed and tool invocation. Ultimately, all of the answers with computational results are summarized into natural language feedback to the user, aligning with the operational requirements of the network and ensuring a professional user experience.

5. LLM for Task Implementation

The operation and maintenance of optical networks involves a multitude of complex tasks that often encompass numerous factors and cannot be simply distilled into a singular problem, such as alarm analysis, fault diagnosis, and physical layer optimization [17,18]. Instead, it usually necessitates the implementation of a systematic technique to solve the task in a step-by-step manner until achieving favorable outcomes. Here, we propose a five-step procedural framework for solving the complex tasks in optical networks based on LLM and LangChain [16], as depicted in Figure 4:

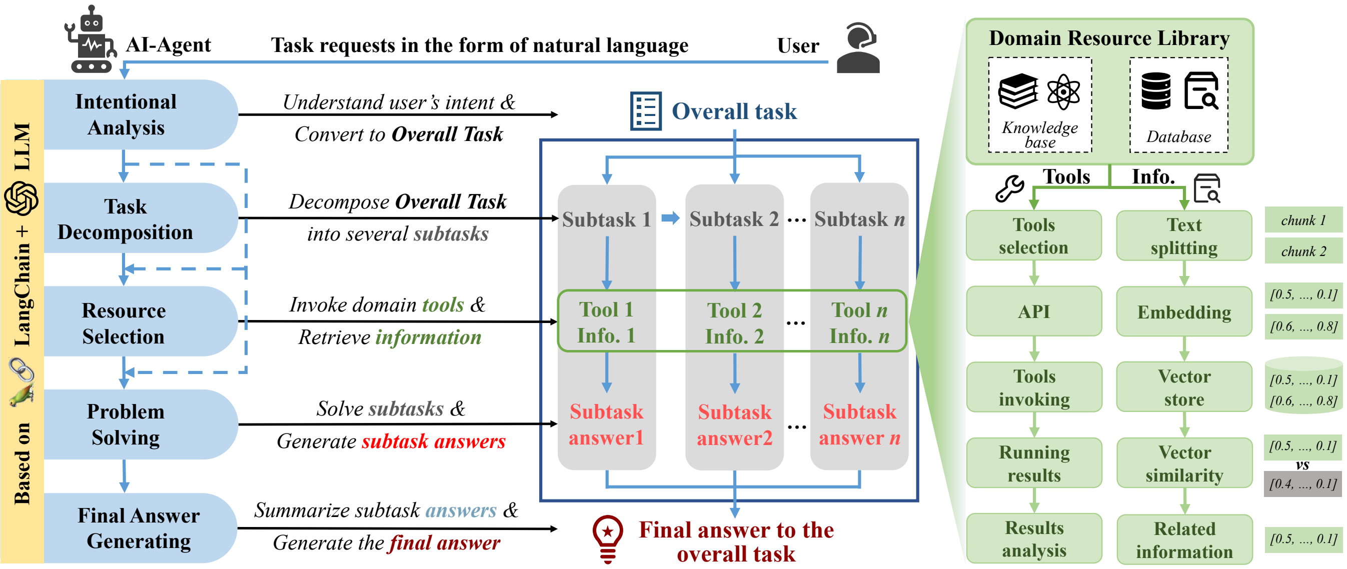


Figure 4. Schematic of procedural framework for solving the complex tasks in optical networks based on LLM and LangChain, including five operating steps: intentional analysis, task decomposition, resource selection, problem solving, and final answer generating.

(1) Intentional analysis: As the first and most critical step in solving complex tasks, the AI-Agent should accurately understand the user's intent and clarify the overall task. These functions are powered by a well-crafted LLM, which has a full understanding and learning of domain tasks and domain resources through mainly few-shot ICL. By learning typical task-solving cases, the agent can provide reliable intent analysis and overall task formulation for the specific task request from the user.

(2) Task decomposition (optional): In most cases, the overall task cannot be answered perfectly in just one step, and it usually needs to be decomposed into several subtasks. Whether the overall task needs to be decomposed, how many subtasks it needs to be split into, and in which order are determined by the well-crafted LLM. With the help of prompt techniques, including CoT and ToT, the LLM could think logically like humans, and learns the chain-solving processes from a series of task-solving cases. For example, in order to analyze and handle the alarms in optical networks, the overall task is decomposed into three subtasks, including alarm compression, alarm correlation analysis, and alarm solving suggestions. By properly solving these three subtasks, the AI-Agent can generate better analysis and suggestions for alarms than those without task decomposition.

(3) Resource selection (optional): Generally, it is tricky for LLMs to solve the diverse task requirements in professional domains, as they often necessitate capabilities beyond NLP. The advanced power of LLMs in professional domains will emerge when they are combined with the domain-specific resources. On the one hand, the LLM can access and invoke domain tools through the API based on its tool selection decisions, and subsequently analyzes the obtained results. On the other hand, the professional information in the domain database can be retrieved quickly and accurately as required, thanks to the RAG technique integrated in LangChain. Specifically, different types of knowledge and data are first loaded and split into smaller chunks. Each chunk is further embedded to capture its semantic meaning so that efficiently identifying similar chunks. The embedded chunks are then stored in the vector store, and a similarity analysis is executed to select the most related chunks for further analysis.

(4) Problem solving: In this step, the LLM will generate a comprehensive answer for each subtask by leveraging the results obtained from the invoked tools, the relevant information retrieved from domain database, and the self-analysis of the LLM pertaining to both current and previous subtasks. The generated answers can effectively solve the current subtasks and provide valuable information for tackling the next subtask. Therefore, similar to the approach employed by experts and experienced engineers in handling complex domain tasks, the LLM is able to generate better results by systematically solving subtasks step by step.

(5) Generating the final answer: Finally, the well-crafted LLM will generate a final answer to the overall task, generally including two patterns: (a) the cascaded pattern, where the outcome of the previous subtask will have an impact on the subsequent one and the final answer is derived from the result of the last subtask; (b) the parallel pattern, where each subtask is independently and simultaneously solved, and the final answer is obtained from a comprehensive summary of all subtasks' results. This approach ensures the final answer is presented with clarity and completeness, thereby enhancing the overall effectiveness of the problem-solving process.

In the process of analyzing and solving numerous problems in optical networks, these five steps can ensure the accuracy, professionalism, and comprehensibility of answers, effectively assisting network operators in tackling complex problems. Note that as supporting technologies, both LLM itself and LangChain are constantly applied throughout the entire process to facilitate effective operation at each step.

6. Applications of LLM in Optical Networks

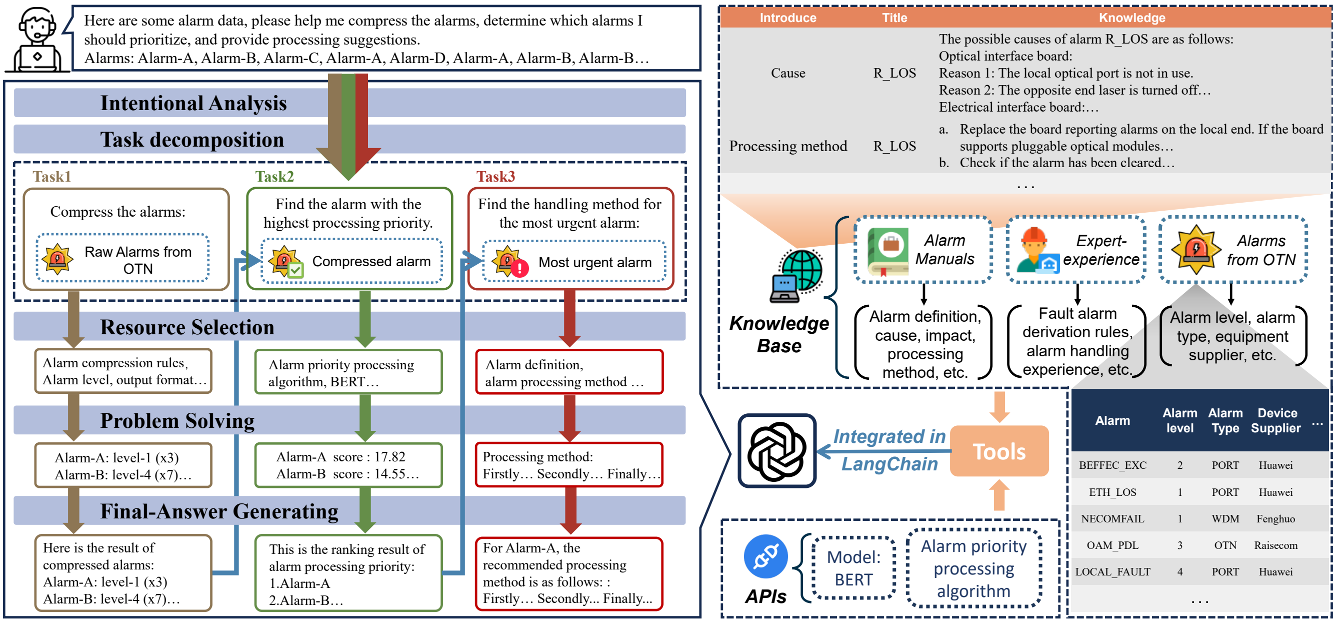
Next, we will apply the LLM to optical networks based on above-introduced techniques. Herein, we study two typical tasks: network alarm analysis and autonomous network optimization. In this section, the up-to-date GPT-4 is selected as the representative LLM for the case study.

6.1. Network Alarm Analysis

Network alarms are the signs that indicate the link status or device incident, such as fault, anomaly, or degradation. Alarm analysis involves the process of detecting, diagnosing, locating, and recovering network issues by analyzing massive alarm data, which is significant for network operators, but still faces challenges in terms of low efficiency, strong reliance on expertise, strenuous manual labor.

We combine network alarm analysis with LLM to automate a series of processes, including alarm compression, priority sorting, and providing alarm handling suggestions [19], as shown in Figure 5. The knowledge extracted from the alarm manual, expert experience, and information from real OTN alarms are integrated into the knowledge base. BERT and alarm priority processing algorithms are integrated into the database. The knowledge base and database are integrated as tools in LangChain, and can be called by the LLM.

Compared to directly querying GPT, our method allows GPT to better understand the principles of alarm analysis and proceed to analyze and solve problems step by step. Then, the overall question is directly provided to GPT-4. After analyzing the operator's intention, the overall question will be divided into three tasks, each of which will be resource selected, task solved, and finally an answer will be generated.



Here are some alarm data, please help me compress the alarms, determine which alarms I should prioritize, and provide processing suggestions.
Alarms: Alarm-A, Alarm-B, Alarm-C, Alarm-A, Alarm-D, Alarm-A, Alarm-B, Alarm-B…

Intentional Analysis
Task decomposition
Task1
Compress the alarms:
Raw Alarms from OTN
Task2
Find the alarm with the highest processing priority.
Compressed alarm
Task3
Find the handling method for the most urgent alarm:
Most urgent alarm
Resource Selection
Alarm compression rules, Alarm level, output format…
Alarm priority processing algorithm, BERT…
Alarm definition, alarm processing method …
Problem Solving
Alarm-A: level-1 (x3) Alarm-B: level-4 (x7)…
Alarm-A score : 17.82 Alarm-B score : 14.55…
Processing method: Firstly… Secondly… Finally…
Final-Answer Generating
Here is the result of compressed alarms: Alarm-A: level-1 (x3) Alarm-B: level-4 (x7)…
This is the ranking result of alarm processing priority: 1.Alarm-A 2.Alarm-B…
For Alarm-A, the recommended processing method is as follows: : Firstly… Secondly... Finally...
Knowledge Base
Alarm Manuals
Expert-experience
Alarms from OTN
Alarm definition, cause, impact, processing method, etc.
Fault alarm derivation rules, alarm handling experience, etc.
Alarm level, alarm type, equipment supplier, etc.
Integrated in LangChain
Tools
APIs
Model: BERT
Alarm priority processing algorithm

Introduce	Title	Knowledge
Cause	R_LOS	The possible causes of alarm R_LOS are as follows: Optical interface board: Reason 1: The local optical port is not in use. Reason 2: The opposite end laser is turned off… Electrical interface board:…
Processing method	R_LOS	a. Replace the board reporting alarms on the local end. If the board supports pluggable optical modules… b. Check if the alarm has been cleared…
		…

Alarm	Alarm level	Alarm Type	Device Supplier	…
BEFFEC_EXC	2	PORT	Huawei	
ETH_LOS	1	PORT	Huawei	
NECOMFAIL	1	WDM	Fenghuo	
OAM_PDL	3	OTN	Raisecom	
LOCAL_FAULT	4	PORT	Huawei	
…				

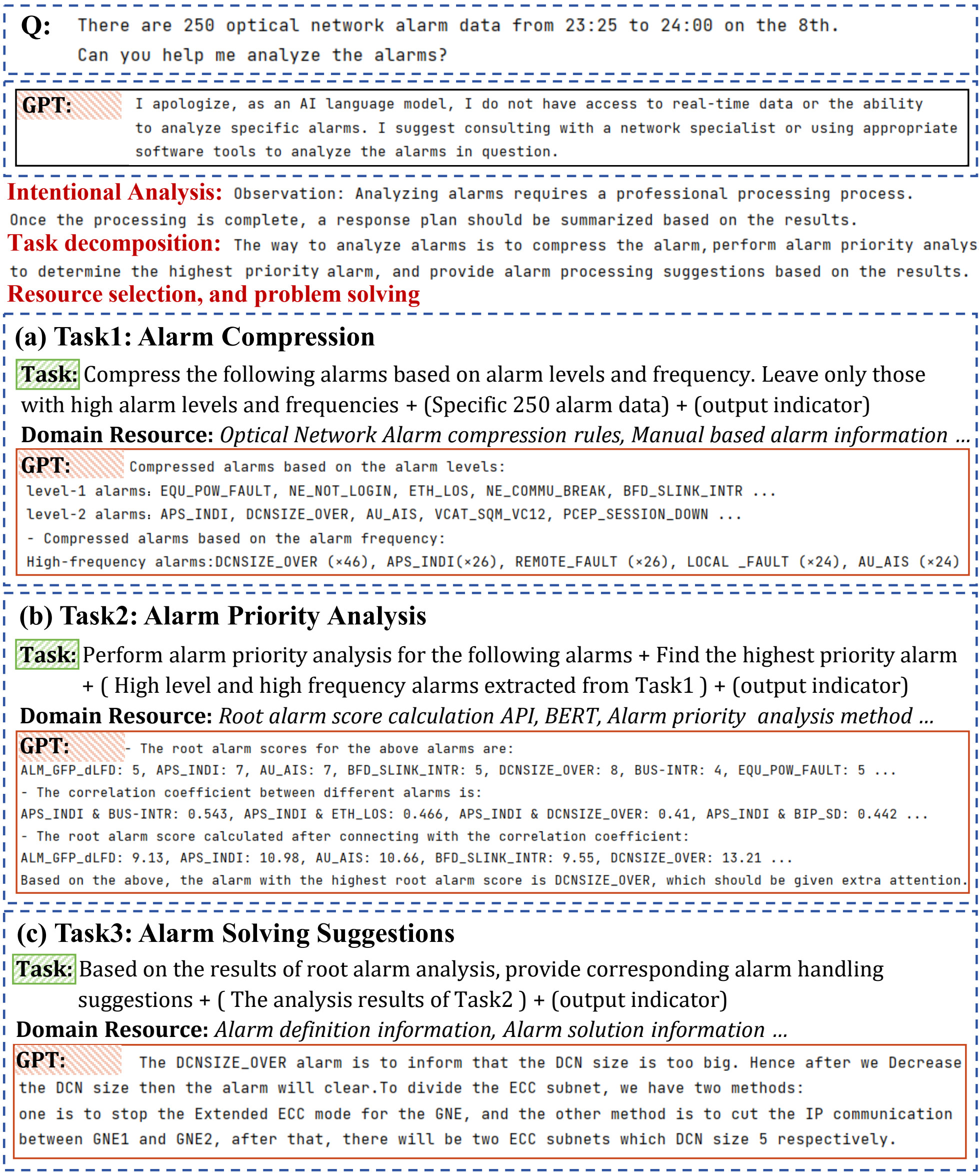
Figure 5. Schematic of LLM-enabled network alarm analysis based on five-step procedural framework, encompassing three tasks: alarm compression, process priority sorting, and alarm solving suggestions.

Task 1: Alarm compression. The urgency of alarms in OTN generally depends on two aspects: the severity level and occurrence frequency. The key to alarm compression lies in these two features, and the compressed alarm can help operators visually observe the overall situation of the alarm event. The OTN alarm information in the knowledge base can help the LLM achieve more accurate alarm compression. The actual effect of alarm compression is shown in Figure 6a.

Task 2: Process priority sorting. This task aims to calculate and update the alarm importance score based on the compression results in Task 1. After the LLM obtains the compressed alarm event information, it obtains the importance score of each alarm by calling BERT and alarm priority processing algorithms. Among them, BERT can obtain the correlation coefficient between different alarms, and the alarm priority processing algorithm combines the alarm level, occurrence frequency, and the correlation coefficient between alarms to calculate the importance score. Then, the LLM can sort the priority of alarm processing based on the results. The actual effect of alarm priority analysis is shown in Figure 6b.

Task 3: Alarm solving suggestions. The alarm with the highest score in Task 2 will be identified as the most urgent alarm of this event. LLM will provide the cause and handling suggestions for this alarm based on the information from the alarm manual and guidance from the alarm processing rule, as the conclusive response generated by AI-Agent. The actual effect of providing alarm suggestions is shown in Figure 6c.

With the help of carefully prepared CoT prompts, LLM is activated to understand and break down the problem, select appropriate knowledge and APIs, and solve the problem step by step. This comprehensive alarm processing scheme showcased the capabilities of LLM, providing a successful solution to this intricate task in optical networks.



Q: There are 250 optical network alarm data from 23:25 to 24:00 on the 8th. Can you help me analyze the alarms?

GPT: I apologize, as an AI language model, I do not have access to real-time data or the ability to analyze specific alarms. I suggest consulting with a network specialist or using appropriate software tools to analyze the alarms in question.

Intentional Analysis: Observation: Analyzing alarms requires a professional processing process. Once the processing is complete, a response plan should be summarized based on the results.

Task decomposition: The way to analyze alarms is to compress the alarm, perform alarm priority analysis to determine the highest priority alarm, and provide alarm processing suggestions based on the results.

Resource selection, and problem solving

(a) Task1: Alarm Compression

Task: Compress the following alarms based on alarm levels and frequency. Leave only those with high alarm levels and frequencies + (Specific 250 alarm data) + (output indicator)

Domain Resource: *Optical Network Alarm compression rules, Manual based alarm information ...*

GPT:
```
Compressed alarms based on the alarm levels:
level-1 alarms： EQU_POW_FAULT, NE_NOT_LOGIN, ETH_LOS, NE_COMMU_BREAK, BFD_SLINK_INTR ...
level-2 alarms： APS_INDI, DCNSIZE_OVER, AU_AIS, VCAT_SQM_VC12, PCEP_SESSION_DOWN ...
- Compressed alarms based on the alarm frequency:
High-frequency alarms:DCNSIZE_OVER (×46), APS_INDI(×26), REMOTE_FAULT (×26), LOCAL _FAULT (×24), AU_AIS (×24)
```

(b) Task2: Alarm Priority Analysis

Task: Perform alarm priority analysis for the following alarms + Find the highest priority alarm + (High level and high frequency alarms extracted from Task1) + (output indicator)

Domain Resource: *Root alarm score calculation API, BERT, Alarm priority analysis method ...*

GPT:
```
- The root alarm scores for the above alarms are:
ALM_GFP_dLFD: 5, APS_INDI: 7, AU_AIS: 7, BFD_SLINK_INTR: 5, DCNSIZE_OVER: 8, BUS-INTR: 4, EQU_POW_FAULT: 5 ...
- The correlation coefficient between different alarms is:
APS_INDI & BUS-INTR: 0.543, APS_INDI & ETH_LOS: 0.466, APS_INDI & DCNSIZE_OVER: 0.41, APS_INDI & BIP_SD: 0.442 ...
- The root alarm score calculated after connecting with the correlation coefficient:
ALM_GFP_dLFD: 9.13, APS_INDI: 10.98, AU_AIS: 10.66, BFD_SLINK_INTR: 9.55, DCNSIZE_OVER: 13.21 ...
Based on the above, the alarm with the highest root alarm score is DCNSIZE_OVER, which should be given extra attention.
```

(c) Task3: Alarm Solving Suggestions

Task: Based on the results of root alarm analysis, provide corresponding alarm handling suggestions + (The analysis results of Task2) + (output indicator)

Domain Resource: *Alarm definition information, Alarm solution information ...*

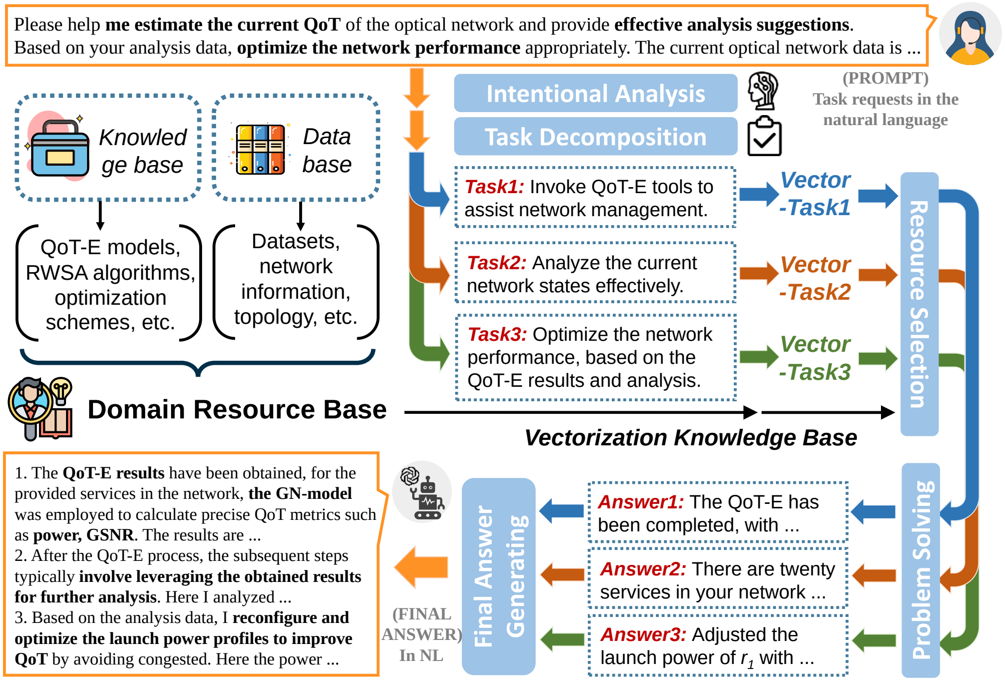
GPT:
```
The DCNSIZE_OVER alarm is to inform that the DCN size is too big. Hence after we Decrease
the DCN size then the alarm will clear.To divide the ECC subnet, we have two methods:
one is to stop the Extended ECC mode for the GNE, and the other method is to cut the IP communication
between GNE1 and GNE2, after that, there will be two ECC subnets which DCN size 5 respectively.
```

Figure 6. Results of network alarm analysis using LLM: (**a**) Alarm Compression: to compress the given alarms based on their alarm-level and frequency of occurrence. (**b**) Alarm Priority Analysis: to identify the highest priority alarm among the compressed alarms. (**c**) Alarm Solving Suggestions: to provide suggestions for handling the highest priority alarm.

6.2. Autonomous Network Optimization

Next, we study another LLM-based scheme for autonomous network optimization with a sequence of subtasks, including QoT estimation, network analysis, resource allocation, and step-by-step performance optimization [20].

We pass on the requests of automated QoT estimation and network performance optimization and its background to GPT-4 through prompts. The network topology information, link physical parameters, and system configuration rules required for the implementation of the tasks are used as knowledge, while network configuration interfaces, computing tools, optimization algorithms, and others are organized into the resource library. The scenario we validated is in the CONUS network with 77 nodes and 99 bidirectional connections, transmitted through the C+L-band. We provided initial information for 15 services distributed throughout the network that require QoT estimations, including the source and destination nodes of the services, launch power and GSNR profiles. When a question is submitted to the AI-Agent, the initial problem is decomposed into three subtasks that are matched with related knowledge from the domain resource library in the form of vectors to be completed in sequence, as shown in Figure 7.



Please help **me estimate the current QoT** of the optical network and provide **effective analysis suggestions**. Based on your analysis data, **optimize the network performance** appropriately. The current optical network data is ...

1. The **QoT-E results** have been obtained, for the provided services in the network, **the GN-model** was employed to calculate precise QoT metrics such as **power, GSNR**. The results are ...
2. After the QoT-E process, the subsequent steps typically **involve leveraging the obtained results for further analysis**. Here I analyzed ...
3. Based on the analysis data, I **reconfigure and optimize the launch power profiles to improve QoT** by avoiding congested. Here the power ...

(PROMPT)
Task requests in the natural language
Intentional Analysis
Task Decomposition
Knowledge base
Data base
QoT-E models, RWSA algorithms, optimization schemes, etc.
Datasets, network information, topology, etc.
Domain Resource Base
Task1: Invoke QoT-E tools to assist network management.
Task2: Analyze the current network states effectively.
Task3: Optimize the network performance, based on the QoT-E results and analysis.
Vector -Task1
Vector -Task2
Vector -Task3
Resource Selection
Vectorization Knowledge Base
Problem Solving
Answer1: The QoT-E has been completed, with ...
Answer2: There are twenty services in your network ...
Answer3: Adjusted the launch power of r_1 with ...
Final Answer Generating
(FINAL ANSWER) In NL

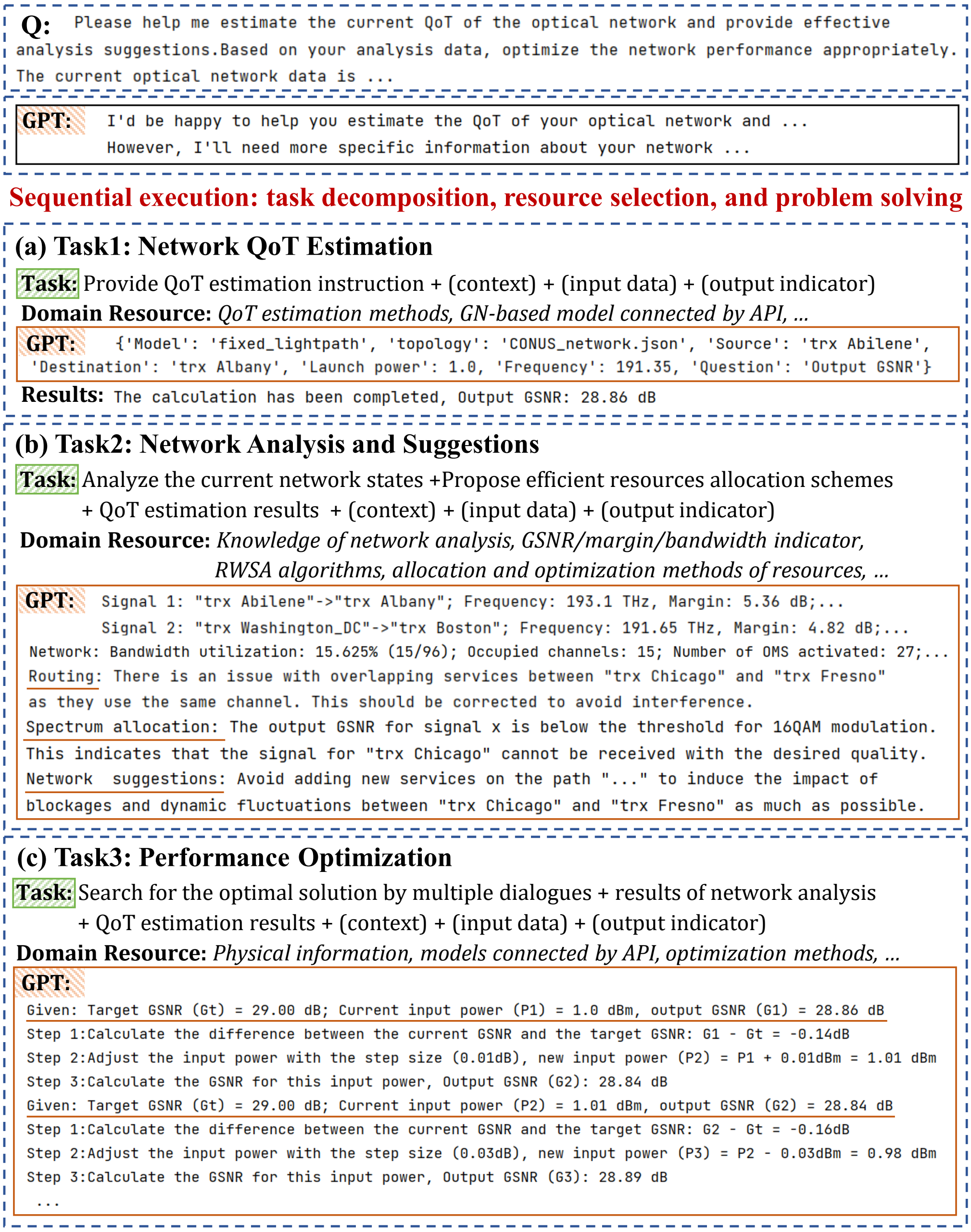
Figure 7. Schematic of LLM-enabled autonomous network optimization based on five-step procedural framework, encompassing three tasks: QoT estimation, network analysis and suggestion, and performance optimization.

Task 1: QoT estimation. This is the process of predicting and evaluating the QoT in terms of GSNR. LLM assists in transferring instructions based on the basic network information such as network topology files, elements configuration files, and parameters monitored; as well as user-provided input parameters, such as signal power, modulation format, and frequency. In this step, natural language in the prompt is transformed into computer instructions, and the GN model [21] in knowledge base are invoked for precise QoT estimation. This prediction serves as a foundation for subsequent network analysis. Figure 8a shows the domain resources provided during the execution of this task. GPT translates our intention into the instructions and we show the final calculation result.

Task 2: Network analysis and suggestions. In this step, the results of QoT estimation and data for network running are utilized as prompt to enable LLM to conduct comprehensive network analysis. The proposed scheme evaluates a range of network metrics, as shown in Figure 8b, including power, GSNR and margin of each channel after each link; blocking probability, spectrum resource utilization and other relevant factors calculated through formulas based on knowledge to comprehensively assess the overall network performance and identify potential parts for improvement.

Task 3: Performance optimization. Based on the resource allocation recommendations, simultaneously in the third step, the LLM guides the iterative optimization process step-by-step. It assists in adjusting launch power profiles and EDFA configuration [21] for the GSNR optimization. This iterative process fine-tunes network parameters, ensuring that optimal transmission conditions are achieved for each connection. Figure 8c shows the output of the iterative process during optimization.

Then, the final answer for the initial question consisting of three subtasks can be obtained from the AI-Agent. The proposed scheme showcases the potential of leveraging LLM in optical networks. By sequentially addressing QoT estimation, network analysis, resource allocation, and GSNR optimization, the LLM can autonomously enhance network performance.



Q: Please help me estimate the current QoT of the optical network and provide effective analysis suggestions.Based on your analysis data, optimize the network performance appropriately. The current optical network data is ...

GPT: I'd be happy to help you estimate the QoT of your optical network and ... However, I'll need more specific information about your network ...

Sequential execution: task decomposition, resource selection, and problem solving

(a) Task1: Network QoT Estimation

Task: Provide QoT estimation instruction + (context) + (input data) + (output indicator)

Domain Resource: *QoT estimation methods, GN-based model connected by API, ...*

GPT:
```
{'Model': 'fixed_lightpath', 'topology': 'CONUS_network.json', 'Source': 'trx Abilene',
'Destination': 'trx Albany', 'Launch power': 1.0, 'Frequency': 191.35, 'Question': 'Output GSNR'}
```

Results: The calculation has been completed, Output GSNR: 28.86 dB

(b) Task2: Network Analysis and Suggestions

Task: Analyze the current network states +Propose efficient resources allocation schemes + QoT estimation results + (context) + (input data) + (output indicator)

Domain Resource: *Knowledge of network analysis, GSNR/margin/bandwidth indicator, RWSA algorithms, allocation and optimization methods of resources, ...*

GPT:
```
Signal 1: "trx Abilene"->"trx Albany"; Frequency: 193.1 THz, Margin: 5.36 dB;...
Signal 2: "trx Washington_DC"->"trx Boston"; Frequency: 191.65 THz, Margin: 4.82 dB;...
Network: Bandwidth utilization: 15.625% (15/96); Occupied channels: 15; Number of OMS activated: 27;...
Routing: There is an issue with overlapping services between "trx Chicago" and "trx Fresno"
as they use the same channel. This should be corrected to avoid interference.
Spectrum allocation: The output GSNR for signal x is below the threshold for 16QAM modulation.
This indicates that the signal for "trx Chicago" cannot be received with the desired quality.
Network  suggestions: Avoid adding new services on the path "..." to induce the impact of
blockages and dynamic fluctuations between "trx Chicago" and "trx Fresno" as much as possible.
```

(c) Task3: Performance Optimization

Task: Search for the optimal solution by multiple dialogues + results of network analysis + QoT estimation results + (context) + (input data) + (output indicator)

Domain Resource: *Physical information, models connected by API, optimization methods, ...*

GPT:
```
Given: Target GSNR (Gt) = 29.00 dB; Current input power (P1) = 1.0 dBm, output GSNR (G1) = 28.86 dB
Step 1:Calculate the difference between the current GSNR and the target GSNR: G1 - Gt = -0.14dB
Step 2:Adjust the input power with the step size (0.01dB), new input power (P2) = P1 + 0.01dBm = 1.01 dBm
Step 3:Calculate the GSNR for this input power, Output GSNR (G2): 28.84 dB
Given: Target GSNR (Gt) = 29.00 dB; Current input power (P2) = 1.01 dBm, output GSNR (G2) = 28.84 dB
Step 1:Calculate the difference between the current GSNR and the target GSNR: G2 - Gt = -0.16dB
Step 2:Adjust the input power with the step size (0.03dB), new input power (P3) = P2 - 0.03dBm = 0.98 dBm
Step 3:Calculate the GSNR for this input power, Output GSNR (G3): 28.89 dB
 ...
```

Figure 8. Results of autonomous network optimization using LLM: (**a**) Network QoT Estimation: to estimate QoT based on network parameters. (**b**) Network Analysis and Suggestions: to analyze the current network state based on the QoT estimation results and propose efficient resource allocation schemes. (**c**) Performance Optimization: to find the optimal solution based on the above analysis results.

6.3. Results and Analysis

In the professional domain, LLMs sometimes suffer from confabulations (or hallucinations), which can result in them making plausible but incorrect statements in some tasks. To study the feasibility and efficacy of LLMs in optical networks, we conducted extensive validations on the two case studies encompassing six tasks, focusing on response accuracy and semantic similarity. A total of 2400 situations were tested (400 situations for each task) under five configuration conditions. More detailed information about the tests used, the evaluation methods, and the processes is presented in Figure 9.

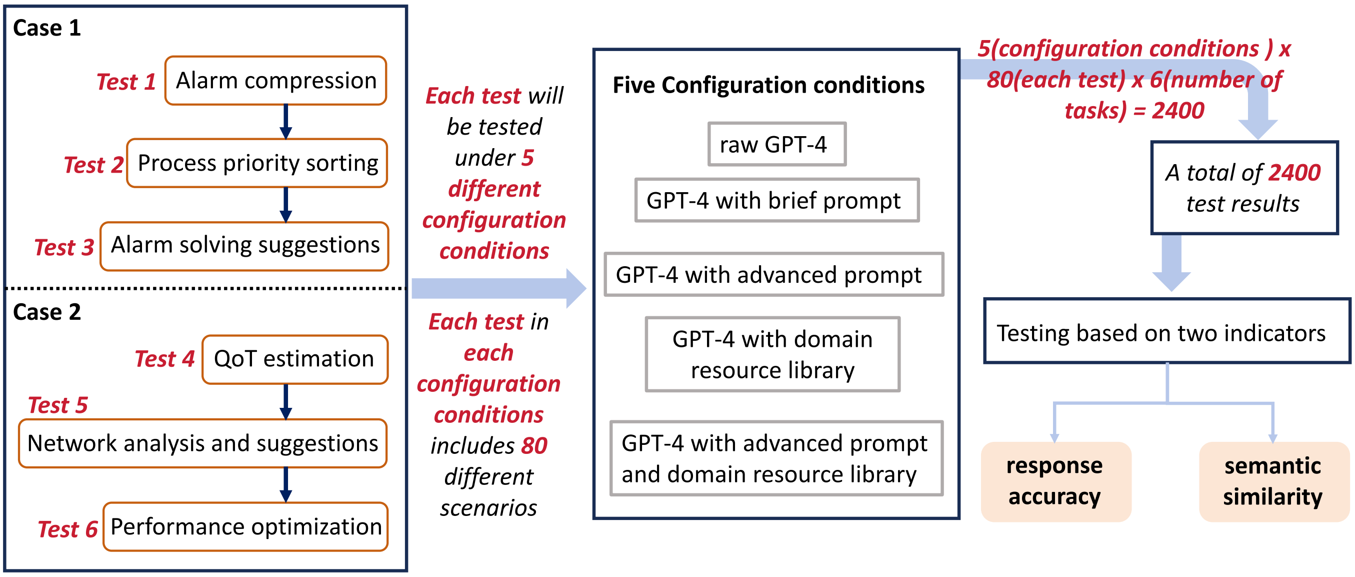

Case 1
Test 1
Alarm compression
Test 2
Process priority sorting
Test 3
Alarm solving suggestions
Case 2
Test 4
QoT estimation
Test 5
Network analysis and suggestions
Test 6
Performance optimization
Each test will be tested under 5 different configuration conditions
Each test in each configuration conditions includes 80 different scenarios
Five Configuration conditions
raw GPT-4
GPT-4 with brief prompt
GPT-4 with advanced prompt
GPT-4 with domain resource library
GPT-4 with advanced prompt and domain resource library
5(configuration conditions) x 80(each test) x 6(number of tasks) = 2400
A total of 2400 test results
Testing based on two indicators
response accuracy
semantic similarity

Figure 9. Implementation flowchart for result testing and analysis.

As shown in Figure 9, the test utilized two cases, each containing three types of test content. For the network alarm analysis case, these included alarm compression, processing priority analysis, and alarm resolution suggestions. For the autonomous network optimization case, these included QoT estimation, network analysis and suggestions, and performance optimization. In total, there are six types of test content, which need to be tested in five different configuration conditions (including raw GPT-4, GPT-4 with brief prompt, GPT-4 with advanced prompt, GPT-4 with domain resource library, and GPT-4 with both advanced prompt and domain resource library, where ICL and CoT are applied in the advanced prompt). Each test includes 80 scenarios in each configuration condition, resulting in a total of 2400 scenarios across all tests. These 2400 test results are evaluated using two metrics: response accuracy and semantic similarity.

In the network alarm analysis test, 2500 out of 29,541 alarms generated in a real OTN within 2 days were adopted, and each 25 adjacent alarms were packaged and randomly selected for testing within a time window of approximately 3 min. In autonomous network optimization, we randomly generate a set of traffic information each time under the same CONUS network topology and link parameters to verify the QoT estimation. Random network state information containing anomalies was provided in the network analysis task to test whether the LLM can accurately analyze insufficient margin, service congestion, conflicts, etc. In the optimization task, we provide random network states to be optimized and verify the optimization task of the network by invoking tools.

First, we evaluated the response accuracy of the AI-Agent on each task under five different configuration conditions to investigate the impact of well-crafted prompts and domain resource library on performance variations in different tasks. Since some open tasks (such as alarm solving suggestions, network analysis and suggestions) are highly subjective, we formulate a specific scoring rule to evaluate the answers generated by LLM. Each answer is scored based on the proportion of key elements it contains, including validity of information, completeness of each step, accuracy of intermediate results, consistence of format, and expert judgment [22]. Under this scoring rule, the accuracies of the LLM on these six tasks under different configuration conditions are calculated, as shown in Figure 10. In all tasks, the raw GPT-4 exhibits the poorest performance, especially in the three tasks of autonomous network optimization that rely more heavily on domain resources. Although only brief prompts are added, the accuracy is improved to a certain extent, and more accurate results can be achieved when advanced prompt is applied, illustrating that well-crafted prompts are significant to improve the LLM's performance. Except for alarm compression, significant accuracy improvements are achieved in other tasks with the help of domain resource library, which illustrates that domain resources are indispensable for domain-specific problems. Particularly, the exception in alarm compression can be primarily attributed to the fact that alarm compression is essentially a traditional NLP problem based on the severity level and occurrence frequency, which requires much fewer domain resources than other tasks. When both the advanced prompt and domain resource library are designed and established, the AI-Agent achieves the optimal accuracy in all tasks, thereby demonstrating the effectiveness of combining prompt engineering and domain resource library for complex tasks in the professional domain.

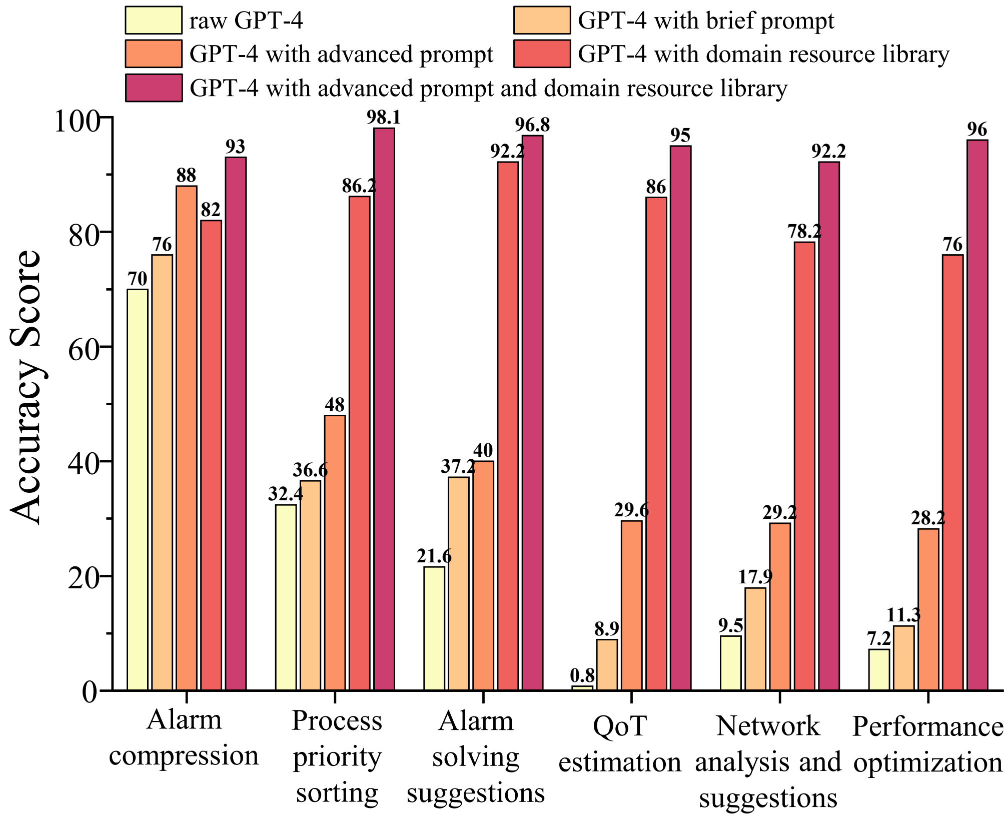


Figure 10. Response accuracy of six tasks for network alarm analysis and autonomous network optimization under five configuration conditions.

Moreover, we analyze the semantic similarity between the answers generated by the LLM and the expected outputs. The purpose of this validation is to verify the effectiveness of the generated results as effective instructions for interaction among application layer, network layer, and physical layer. The SentenceTransformer [23] is applied as the evaluation model, because it is capable of generating meaningful embeddings that reflect semantic content, enabling semantic similarity assessment to ensure reliability and reproducibility. Similarly to the accuracy validation experiment, the three subtasks of network alarm analysis achieve higher semantic similarity than the tasks in autonomous network optimization when only prompt engineering is applied, as shown in Figure 11. Compared with only applying the prompt engineering, when only the domain resource library is applied, the performance optimization task achieves a great improvement, while the alarm solving suggestions task even experienced a decrease of 2%. After integrating the advanced prompt with the domain resource library, the similarity of all six tasks is significantly improved than utilizing only one of them, proving that the combination of these two technologies plays a pivotal role in standardizing the responses generated by the LLM.

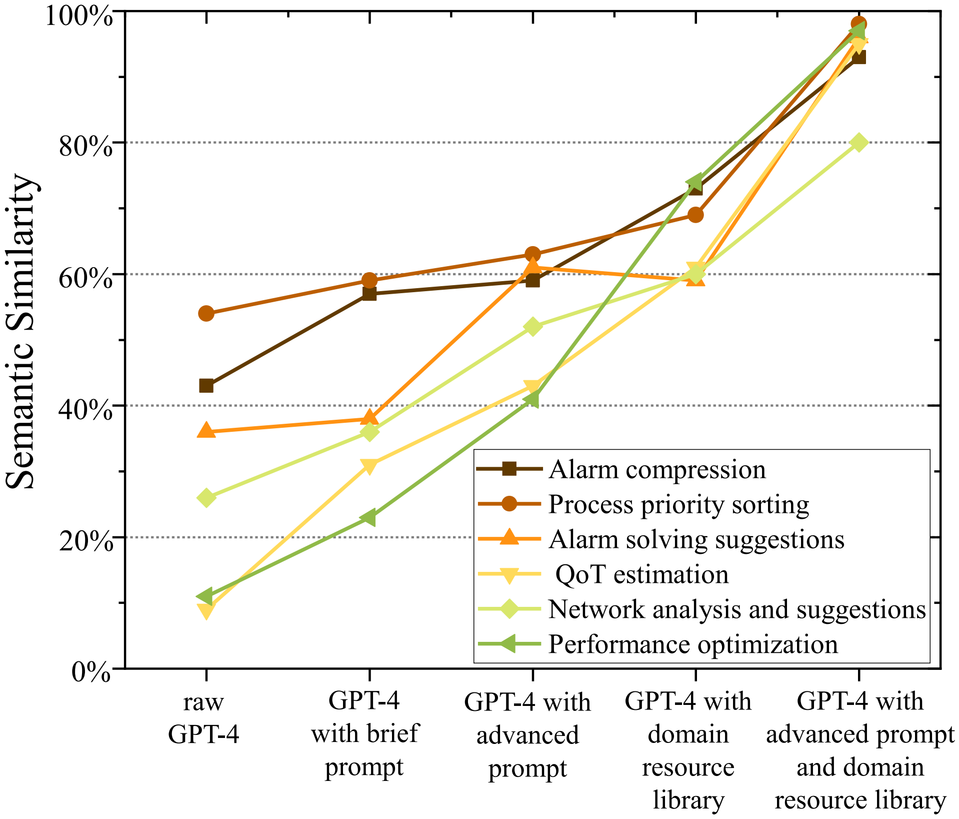


Figure 11. Semantic similarity of six tasks in network alarm analysis and autonomous network optimization under five configuration conditions.

In the optical network framework empowered by LLMs, despite its significant advantages in intelligent physical layer control and efficient interaction with the application layer, potential ethical challenges and security risks must be addressed. Therefore, we propose several solutions to enhance the system's security and reliability. First, automatic monitoring and verification modules, as well as human evaluation, can be implemented to assess the quality of LLM-generated responses. Low-quality or potentially harmful responses should be discarded to ensure the safe operation of the network. Additionally, a reliable logging system to record operations and outputs at each stage is necessary, as it can effectively help operators quickly identify the root cause of issues, improving the accuracy of commands and network security. Lastly, the access and operation permissions of the LLM on the network can be restricted. Considering network security, for operations that may significantly impact the physical layer or system, the LLM's outputs should serve as recommendations for operators rather than direct commands. Manual verification and review must be conducted before any instructions are issued by the LLM to further ensure their correctness. Network administrators should check the accuracy and applicability of the instructions based on their operational experience to prevent unexpected interruptions or failures. Through the above measures, we can effectively reduce the potential security risks and ethical issues while fully leveraging the advantages brought by LLM technology, ensuring the safe and reliable operation of optical networks.

7. Conclusions

In this study, we proposed a framework of LLM-empowered optical networks that enabled intelligent control of the physical layer and efficient interaction with the application layer through an AI-Agent deployed in the control layer. According to the user's input and well-crafted prompts, the AI-Agent can accordingly invoke external tools and extract domain knowledge from the established domain resource library to generate control instructions and result representations, thereby realizing intelligent network operation and maintenance. We extensively deliberated on strategies for proficient prompt engineering, establishment of a comprehensive domain resource library, and successful implementation of intricate tasks, all of which are indispensable for enhancing LLM's proficiency in professional domains and unlocking its potential in tackling complex assignments. Moreover, the applications of LLM in network alarm analysis and autonomous network optimization were also studied. The results showed that LLM exhibited expertise in optical networking. Predictably, LLMs will play an increasingly important role in optical networks in the future, fundamentally transforming the traditional operation and maintenance paradigm.

Author Contributions: Conceptualization, D.W.; methodology, D.W., Y.W., X.J. and Y.Z.; software, Y.W., X.J. and Y.Z.; validation, Y.W., X.J. and Y.Z.; formal analysis, Y.W., X.J. and Y.Z.; investigation, Y.W., X.J. and Y.Z.; resources, Y.W., X.J. and Y.Z.; data curation, Y.W., X.J. and Y.Z.; writing—original draft preparation, D.W., Y.W., X.J. and Y.Z.; writing—review and editing, D.W., Y.W., X.J., Y.Z. and Y.P.; visualization, D.W., Y.W., X.J. and Y.Z.; supervision, D.W., Y.W., X.J., Y.Z. and Y.P.; project administration, M.Z.; funding acquisition, D.W. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 62171053, and Beijing Nova Program grant number 20230484331.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data underlying the results presented in this paper are not publicly available at this time, but may be obtained from the authors upon reasonable request.

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

References

1. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
3. Sanderson, K. Gpt-4 is here: What scientists think. *Nature* **2023**, *615*, 773.
4. Lee, P.; Bubeck, S.; Petro, J. Beneffts, limits, and risks of gpt-4 as an ai chatbot for medicine. *New Engl. J. Med.* **2023**, *388*, 1233–1239.
5. Musumeci, F.; Rottondi, C.; Corani, G.; Shahkarami, S.; Cugini, F.; Tornatore, M. A tutorial on machine learning for failure management in optical networks. *J. Light. Technol.* **2019**, *37*, 4125–4139.
6. Jiang, X.; Zhang, M.; Song, Y.; Zhang, Y.; Wang, Y.; Ju, C.; Wang, D. OptiComm-GPT: A GPT-based versatile research assistant for optical fiber communication systems. *Opt. Express* **2024**, *32*, 20776–20796.
7. Wang, D.; Song, Y.; Zhang, Y.; Jiang, X.; Dong, J.; Khan, F.N.; Sasai, T.; Huang, S.; Lau, A.P.T.; Tornatore, M.; et al. Digital Twin of Optical Networks: A Review of Recent Advances and Future Trends. DOI: 10.1109/JLT.2024.3401419. *J. Light. Technol.* **2024**.
8. Musumeci, F.; Rottondi, C.; Nag, A.; Macaluso, I.; Zibar, D.; Rufffni, M.; Tornatore, M. An overview on application of machine learning techniques in optical networks. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1383–1408.
9. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv* **2022**, arXiv:2206.07682.
10. Patrick, L.; Ethan, P.; Aleks, ra, P.; Fabio, P.; Vladimir, K.; Naman, G.; Heinrich, K.; Mike, L.; Wen-tau, Y.; Tim, R.; Sebastian, R.; Douwe, K. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9459–9474.
11. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35.
12. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, We.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledgeintensive nlp tasks. *Adv. Neural Inf. Process. Syst.* 2020, 33, 9459–9474.
13. Qingxiu, D.; Lei, L.; Damai, D.; Ce, Z.; Zhiyong, W.; Baobao, C.; Xu, S.; Jingjing, X.; Lei, L.; Zhifang, S. A survey on in-context learning. *arXiv* **2022**, arXiv:2301.00234.
14. Jason, W.; Xuezhi, W.; Dale, S.; Maarten, B.; brian, i Fei, X.; Ed, C.; Quoc, V.L.; Denny, Z. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.

15. Wang, D.; Zhang, Z.; Zhang, M.; Fu, M.; Li, J.; Cai, S.; Zhang, C.; Chen, X. The role of digital twin in optical communication: Fault management, hardware conffguration, and transmission simulation. *IEEE Commun. Mag.* **2021**, *59*, 133–139.
16. Topsakal, O.; Akinci, T.C. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. *Int. Conf. Appl. Eng. Nat. Sci.* **2023**, *1*, 1050–1056.
17. Jia, J.; Wang, D.; Zhang, C.; Yang, H.; Guan, L.; Chen, X.; Zhang, M. Transformer-based alarm context vectorization representation for reliable alarm root cause identification in optical networks. In Proceedings of the 2021 European Conference on Optical Communication (ECOC), Bordeaux, France, 12–16 September 2021; pp. 1–4.
18. Wang, C.; Yoshikane, N.; Elson, D.; Tsuritani, T. Automation of fast configuration error diagnosis in optical transport networks–natural language processing is all you need. In *Optical Fiber Communication Conference*; Optica Publishing Group: San Diego, CA, USA, 2023; p. M3G-6.
19. Wang, Y.; Li, J.; Pang, Y.; Song, Y.; Zhang, L.; Zhang, M.; Wang, D. AlarmGPT: An intelligent operation assistant for optical network alarm analysis using ChatGPT. In Proceedings of the 49th European Conference on Optical Communications (ECOC 2023), Glasgow, UK, 1–5 October 2023; Volume 2023, pp. 1330–1333.
20. Zhang, Y.; Zhang, M.; Yuchen, S.; Xiaotian, J.; Yidi, W.; Wang, D.; Shen, S. GPT-enabled digital twin assistant for multi-task cooperative management in autonomous optical network. In *Optical Fiber Communication Conference*; Optica Publishing Group: San Diego, CA, USA, 2024; p. Th1G-4.
21. Song, Y.; Fan, Q.; Lu, C.; Wang, D.; Lau, A.P.T. Efffcient three-step ampliffer conffguration algorithm for dynamic c+ l-band links in presence of stimulated raman scattering. *J. Light. Technol.* **2022**, *41*, 1445–1453.
22. Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv* **2023**, arXiv:2302.04023.
23. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv* **2019**, arXiv:1908.10084.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.