LLM Fine-Tuning for Enhanced Network Understanding: A Rephrase and Contrast Approach

Liujianfu Wang*, Jingqi Lin*, Yuyang Du*, *Graduate Student Member, IEEE*, Kexin Chen, Soung Chang Liew *Fellow, IEEE*

***Abstract*—Large language models (LLMs) are being widely explored for potential applications across diverse disciplines, with significant recent efforts focusing on adapting them to understand how communication networks operate. However, over-reliance on prompting techniques hinders the exploitation of the generalization ability of these models, and the lack of efficient fine-tuning methods prevents the full realization of lightweight LLMs' potential. This paper addresses these challenges by introducing an efficient fine-tuning framework referred to as Rephrase and Contrast (RaC). RaC enhances LLMs' comprehensions and reasoning abilities by incorporating question reformulation and contrastive analysis of correct and incorrect answers into the fine-tuning process. Experimental results of the fine-tuned model demonstrate a 15.84% accuracy improvement over the foundational model when tested on a comprehensive networking problem set. Moreover, for efficient construction of the fine-tuning dataset, we develop a GPT-assisted data mining method for generating high-quality question-answer (QA) pairs. Further, we introduce ChoiceBoost-X, a data augmentation technique that expands dataset size while reducing answer-order bias. We open-source our contributions to the community, including: 1) a networking QA model fine-tuned on LLaMA3.1-8B, 2) training data mined from networking textbooks, 3) testing problem sets of different difficulties, which could serve as benchmark baselines for future research.**

***Index Terms*—Large language model, wireless/wired networks supervised fine-tuning, benchmark dataset**

I. Introduction

In recent years, generative AIs have undergone a paradigm shift with the emergence of large language models (LLMs) that demonstrate near-human intelligence across diverse domains. LLMs have sparked widespread interest on their potential applications in numerous disciplines.

The domain of communication networks is no exception to this trend. Recent research has explored the application of LLMs to several critical areas within the networking field. For example, investigations have demonstrated the potential of LLMs in network topology analysis [1], [2], wireless device configuration [3], [4], network setup optimization [5]–[7], network diagnosis [8], [9], and network security [10]. These studies highlight LLMs' transformative potential in enhancing the performance and efficiency of various networking tasks.

Despite the growing interest and progress in this area, existing approaches for integrating LLMs into communication networks face significant limitations. One prominent limitation stems from the use of prompt-tuning techniques. Although prompt-tuning methods, such as chain of thought (CoT), enable rapid proof-of-concept implementations, they are inherently inefficient and often cumbersome. Designing effective prompts typically requires extensive trial and error, and the resulting solutions may lack robustness and generalizability because prompting does not adapt the parameters within LLMs for network specific tasks. Additionally, the computational overhead associated with prompt-tuning can be substantial, particularly for large-scale network applications [11].

Fine-tuning the parameters of LLMs for networking tasks presents a viable solution to address the limitations of prompt engineering. In this paper, we propose an efficient LLM fine-tuning framework termed Rephrase and Contrast (RaC). Unlike traditional fine-tuning methods that rely solely on question-answer (QA) pairs, RaC introduces additional context to the LLM, including: 1) a reformulation of the original question and 2) a contrastive analysis of both correct answers and plausible but incorrect ones. This approach, inspired by human learning processes, enhances the LLM's comprehension of the underlying problems and significantly improves its critical reasoning capabilities, which are essential for addressing complex networking challenges. Experimental evaluations reveal that the RaC framework achieves a 15.84% improvement in accuracy compared to the baseline model without fine-tuning, as measured on a comprehensive networking problem set.

In addition to the RaC framework, we address the scarcity of datasets for training LLMs on networking tasks. To close this gap, we introduce a GPT-assisted data mining strategy and propose ChoiceBoost-X, an effective data augmentation technique, for constructing training datasets.

For data mining, we leverage the language processing capabilities of GPT-4 to extract and analyze raw material from networking textbooks, generating high-quality QA pairs tailored to the needs of RaC fine-tuning. The QA generation process employs in-context learning (ICL) to guide GPT-4 in structuring the data and producing outputs aligned with the requirements of each step.

For data augmentation, our ChoiceBoost-X technique enhances the dataset by rearranging the order of correct and incorrect answers in multiple-choice questions. This method expands the dataset by a factor of twenty-four while preserving the original knowledge content. By varying the answer arrangements, ChoiceBoost-X reduces biases induced by answer order and strengthens the model's understanding of the material. Ablation studies confirm that the inclusion of ChoiceBoost-X results in measurable performance improvements for the fine-

This work was supported in part by the Shen Zhen-Hong Kong-Macao technical program (Type C) under Grant No. SGDX20230821094359004.

L. Wang, Y. Du, K. Chen, and S. C. Liew are with the Chinese University of Hong Kong (e-mail: {wl024, dy020, soung}@ie.cuhk.edu.hk, kxchen@cse.cuhk.edu.hk). J. Lin is with Huazhong University of Science and Technology (email: linjingqi0613@hust.edu.cn). S. C. Liew is the corresponding author.

*L. Wang, J. Lin, and Y. Du contribute equally to this work. The work was completed during J. Lin's internship at the Chinese University of Hong Kong.

An early version of this paper has been presented in IEEE ICNC 2025.

tuned model.

Beyond our technical contributions – namely, the RaC fine-tuning framework, the GPT-assisted data mining approach, and ChoiceBoost-X – we also provide several resources to advance research in network-oriented LLMs. These resources include: 1) the training dataset developed for LLM fine-tuning, 2) three novel testing problem sets designed as benchmarks for future studies, and 3) a networking QA model fine-tuned on LLaMA3.1-7B. By releasing these contributions, we aim to enable researchers and practitioners to reproduce our results and build upon our work, thereby fostering the development of advanced LLMs tailored for communication networks.

Remark: A preliminary version of this work was presented at IEEE ICNC 2025 [12]. This paper is a significant extension and improvement of the previous conference version. First, as a replacement for the Llama2-7B model previously used, a more advanced Llama3.1-8B model is applied as the foundational model, enabling better performance after fine-tuning. Second, we enlarge the training dataset by mining additional networking textbooks. To accommodate the wider range of knowledge covered in the training data, we have also expanded the testing dataset with more manually crafted questions. Third, we develop ChoiceBoost-X, an efficient data augmentation method that increased the size of the training data by a factor of 24, replacing the less efficient method previously reported at the conference. Finally, new experiments have been conducted to validate the system. The project is available at: https://github.com/1155157110/RaC_open_dataset.

II. Related Work

Recent advances in LLMs have inspired many researchers to explore their use for networking tasks [1]–[10], [13], [14]. This section highlights this paper's unique contributions compared with these previous works. Our comparison centers on two key aspects: 1) the methodologies employed, and 2) the availability of components associated with these projects, including codes, training data, and testing problem sets.

Methodologies: Most previous works were built upon prompt tuning over off-the-shelf models with only a few considered model fine-tuning. For example, in [7], the authors put forth a framework that empowers LLMs with various model-optimizing techniques including model fine-tuning. However, the work only conducted experiments on prompt-tuning methods. In [1], [3]–[5], [7]–[10], few-shot prompting methods, such as ICL, are used to guide LLMs to generate answers reliably. CoT was applied in [1], [5], [7], [9] to enhance LLMs' reasoning ability. Additionally, retrieval methods like retrieval augmented generation (RAG) are used in [7], [13] to enhance the off-the-shelf LLM's knowledge in wireless networks.

These off-the-shelf models were not specifically trained for networking applications. Their expertise is in conventional NLP tasks, such as language translation, rather than in networking. Despite the advanced prompting techniques to enhance LLMs' comprehension of networks, their efficacy is limited due to the inherent limitations in LLMs' foundational architectures.

Within the small handful of works that fine-tuned LLMs for wireless networking tasks, [6], [15] adapted a LLaMA2 model with the low-rank adaptation (LoRA) technique and demonstrated the model's performance in specific tasks such as adaptive bit rate streaming and cluster job scheduling. However, these paper did not make new contributions in fine-tuning methods: they used the LoRA method without further modification to enhance its performance.

Unlike the above previous work, our paper put forth the RaC scheme for effective LLM fine-tuning (see Section III for details). Furthermore, to complement, we develop a GPT-assisted data mining method and a data augmentation scheme to gloom the dataset used to fine-tune LLMs (see Section IV-A for details). These methods make possible a more efficient utilization of the knowledge within the training dataset.

Project Components: In previous work, only [2], [4], [13], and [14] have made their data available online. However, the released datasets are either too small for LLM adaptations [2], [4], or cannot be directly used in model fine-tuning, as the corresponding code for data post-processing and scripts for fine-tuning models are not released [13], [14]. By contrast, we provide a self-contained dataset for LLM adaptation. Additionally, we also provide the community with the source code associated with the model fine-tuning, facilitating reproductions and follow-ups of this work. Moreover, having recognized the absence of a generally accepted testing benchmark for evaluating LLMs' performance on networking tasks, we release three testing problem sets of different difficulties to serve as network-oriented LLM evaluation baselines.

III. RaC: An Efficient Fine-tuning Framework

The design of RaC is inspired by the following observations:

Observation 1 (the importance of rephrasing): In human society, question rephrasing plays a crucial role in ensuring accurate and efficient communication. When faced with an unfamiliar question, we often rephrase it to eliminate ambiguities. Additionally, providing detailed context for the question enhances the responder's understanding.

Observation 2 (wrong answers matter): In human learning, analyzing incorrect answers enhances understanding. By comparing them with correct answers, we gain deeper insights into the topic, fostering critical thinking. Moreover, explanations of incorrect answers can introduce new information that may not be covered in the explanation of the correct answer.

Building on these observations, our RaC framework supplements the data used to fine-tune LLMs by rephrasing questions and performing contrastive analysis of correct and incorrect answers. Fig. 1 illustrates this process with an example.

Before delving into Fig. 1, we briefly explain the rationale for choosing multiple-choice QA format. First, this format offers an objective and easy-to-grade method for evaluating the model's accuracy. Additionally, it allows for analysis of the model's decision-making process by assessing its ability to identify the best answer among multiple plausible options.

In Fig. 1, the multiple-choice question asks about the intended function of the STARTTLS command, commonly used in the Simple Mail Transfer Protocol (SMTP). The correct answer is A) It encrypts the entire SMTP session.

Using the RaC method, we first rephrase the question by clarifying the term "intended function" and pointing out that

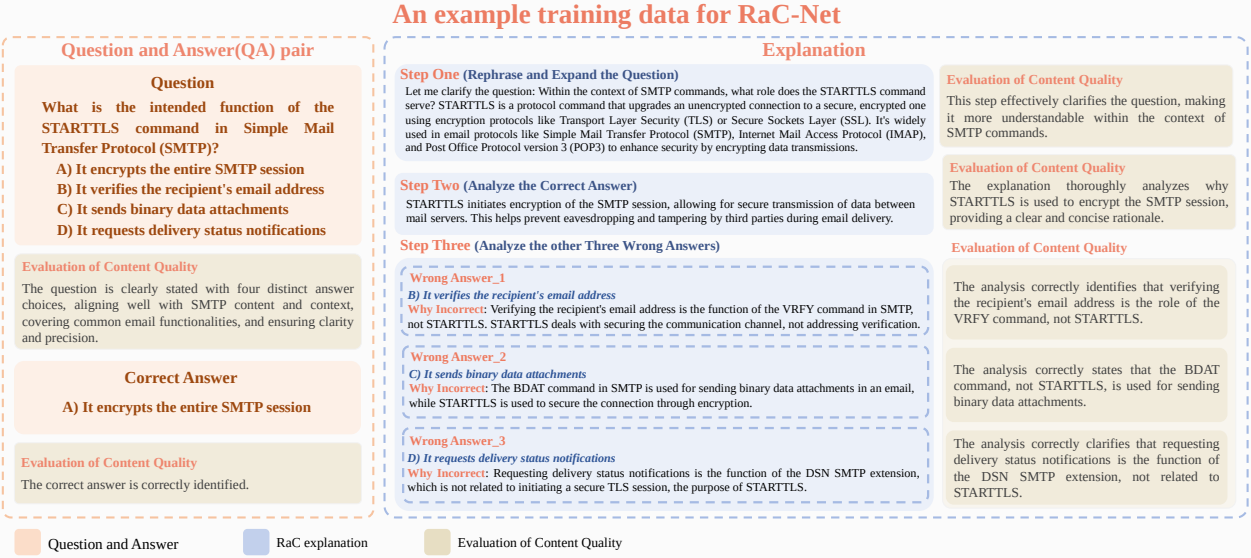

Figure 1. An example of training data that meets the data requirements of our RaC framework. For easier understanding and better assessment of data quality, comments have been added in dark yellow boxes. These comments are not part of the training dataset; they are included here for illustrative purposes only.

STARTTLS is a command that upgrades an existing, unencrypted connection to a secure, encrypted one using TLS or SSL. It is widely used in email protocols SMTP, IMAP, and POP3 to enhance security by encrypting data transmissions.

We then move on to contrastive analysis of the correct and incorrect answers. We explain that the correct answer identifies STARTTLS as the command that encrypts the SMTP session, emphasizing its role in securing email transmissions by preventing third-party eavesdropping and tampering. In contrast, each incorrect answer mistakenly assigns the functions of other SMTP commands to STARTTLS, such as VRFY for verifying email addresses, BDAT for sending binary data, and DSN for requesting delivery status. This analysis effectively differentiates between various SMTP commands, highlighting STARTTLS's specific security role. It is important to note that all of the question reformulation contents and contrastive analysis are autonomously generated by the GPT model, without human intervention (see Section IV for details).

For the RaC implementation, our framework can be adopted in a variety of supervised fine-tuning methods. We implement LoRA to verify the performance of our RaC framework. Compared with LoRA without RaC, our RaC framework improves models' comprehension in the fine-tuning process.

IV. Training and Testing Datasets

The extensive dataset required for LLM fine-tuning necessitates an automated approach for QA pair generation. We leverage the GPT-4 model to generate a substantial corpus of high-quality QA pairs. Importantly, we emphasize that the QA pairs of interest to us are more than a simple combination of a question and an answer. A core feature of our QA pairs is the inclusion of problem rephrasing and contrasting statements, which are required in the RaC framework. The rest of this paper refers to the new QA pair as the RaC QA pair to distinguish it from the conventional QA pair.

In the following, Subsection A presents technical details about the data generation process, including our GPT-assisted data mining scheme and ChoiceBoost-X, the permutation-based data augmentation method designed for the multiple-choice setup in RaC. Subsection B introduces our open-sourced training dataset and testing problem sets for future research endeavors.

A. Dataset Construction

Our dataset is built upon 12 textbooks on computer networking, including both foundational theories and the latest technical advancements in the networking field. Each textbook was chosen for its unique technical insights to ensure a comprehensive coverage of the networking field. A detailed list of these books is provided in the project's github page.

In our dataset, each instance is a multiple-choice QA pair, including one correct answer, three incorrect answers, problem rephrase, and explanations associated with these four choices. This data structure aligns with the RaC fine-tuning framework. The following procedure was applied to ensure an efficient and high-quality data mining process.

Preprocessing: We employed optical character recognition (OCR) to convert the selected textbooks into textual material. Along with the OCR process, we removed elements incompatible with the GPT-4 model, such as images and their captions. The content was automatically segmented based on textbook subsections to preserve contextual coherence. Consecutive subsections were then merged, provided that the combined length does not exceed the predefined token limit of GPT-4. After purification and segmentation, the resulting textual context of each book section was structured into the JSON format for subsequent processing.

RaC QA Pair Generation: We leveraged the GPT-4 API for QA pair generations, with the following configuration parameters used: temperature = 1.0, top-p = 1.0, frequency penalty = 0.0, and presence penalty = 0.0. Fig. 2 illustrates the detailed prompt employed in QA pair generation. The process begins by describing the basic task for the model: the creation

of questions targeting networking knowledge. Then the prompt outlines the requirements for question content and structure. Finally, it defines a three-step strategy for generating the RaC explanations.

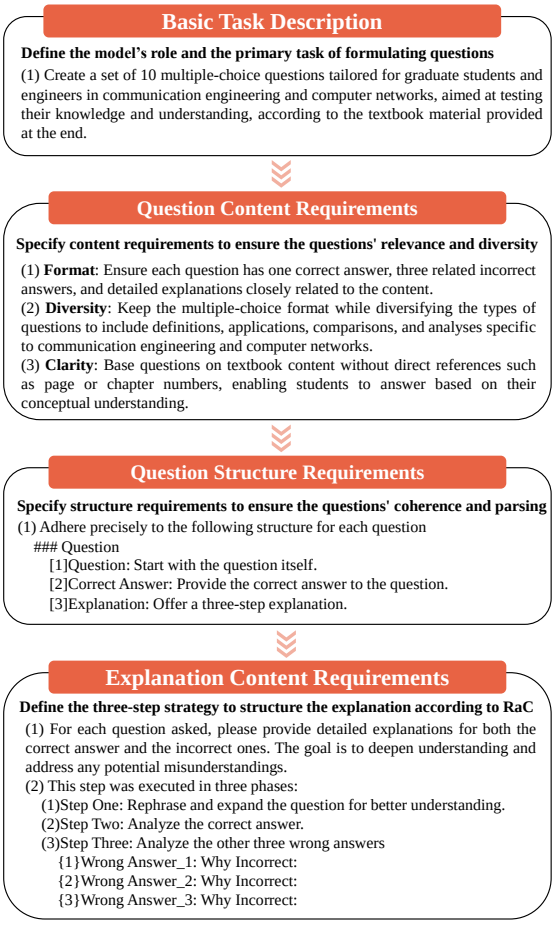

Figure 2. LLM prompt template used for creating QA pairs.

Manual Review: Following QA pair generation, we randomly sampled 200 QA pairs for evaluation. Each selected QA pair underwent a manual review. If any component (including question, answer, or explanation) was found to be inaccurate or illogical, we returned to the prompt design and revised it accordingly. This iterative process continued until the dataset met our stringent quality criteria, i.e., meticulously crafted questions, exact answers, and logically coherent, structurally sound explanations.

We next explain how we augment the obtained dataset with ChoiceBoost-X. Analysis of the generated data reveals two significant observations: 1) While GPT-4's assistance facilitates the production of high-quality data, the associated API costs would be prohibitive for large-scale data expansion. 2) The generated data exhibited potential bias in option selection, which could mislead the fine-tuned model into acquiring unfounded statistical bias.

We design ChoiceBoost-X to mitigate the above two issues. In ChoiceBoost-X, each question in the dataset was duplicated twenty-four times, with the options arranged according to all twenty-four permutations. Without additional API cost, this approach substantially expands the dataset (which enhances the model's capacity by learning from diverse choice setup) and mitigates biases inherent in the initially generated data.

Note that ChoiceBoost-X is a significant improvement over ChoiceBoost, the data augmentation technique reported in our previous conference paper [12]. ChoiceBoost is designed to address biases by duplicating the training data four times, with each duplicate assigned a different correct answer option (A, B, C, or D), while preserving the original order of the incorrect options. For example, marking an asterisk by the correct answer, {A*,B,C,D} becomes {A*,B,C,D}, {A,B*,C,D}, {A,B,C*,D}, and {A,B,C,D*}. This approach eliminates the bias introduced by the fixed ordering of correct options. However, a limitation remains: the incorrect options exhibit repeated patterns across the duplicates, with each incorrect answer remaining consistently associated with the same option in three out of the four duplicates. To address this issue and further reduce the correlation between incorrect options and their corresponding incorrect answers, we put forth ChoiceBoost-X, an optimized version of the technique. ChoiceBoost-X fully decouples the options from the content of the answers, thereby eliminating residual patterns in the ordering of incorrect answers. Section V presents ablation experiments that demonstrate the superior learning performance achieved by ChoiceBoost-X.

B. Data Resources Released

The initial dataset derived from twelve textbooks without post-processing data augmentation comprises 17,176 QA pairs. These original QA pairs encompass essential knowledge of computer networking subdomains described in each individual textbook, such as layered network architecture, protocol design, network security, and network management. From the raw data, 1,718 QA pairs (approximately 10%) were randomly selected to form a testing problem set. We then used ChoiceBoost-X to augment the remaining 17,176 – 1,718 = 15,458 QA pairs, resulting in an unbiased training dataset containing 15,458 * 24 = 370,992 QA pairs.

In addition to the GPT-generated QA pairs, we manually collected 401 multiple-choice testing questions closely aligned with the content of the twelve selected textbooks. These questions were sourced from publicly available resources, including open-source exam papers in online databases and problem sets provided within the textbooks. Note that these QA pairs are exclusively used for testing purposes only, and they lack problem rephrasing and answer explanations required in the RaC framework. The value of the new testing problem set lies in its focus on complex logical reasoning and intricate calculations. For instance, such problems might require calculating the maximum network throughput under changing bandwidth conditions, or involve multi-step reasoning, such as optimizing a routing algorithm considering traffic fluctuations and delay constraints. The motivation for building the more challenging dataset stems from our observation that GPT-generated QA pairs may not adequately address complex reasoning and mathematical computations due to the limitations of the GPT-4 model in these areas. Instead, the GPT-generated pairs primarily emphasize concept comprehension. For example, a typical problem from the set might ask, "What is the function of

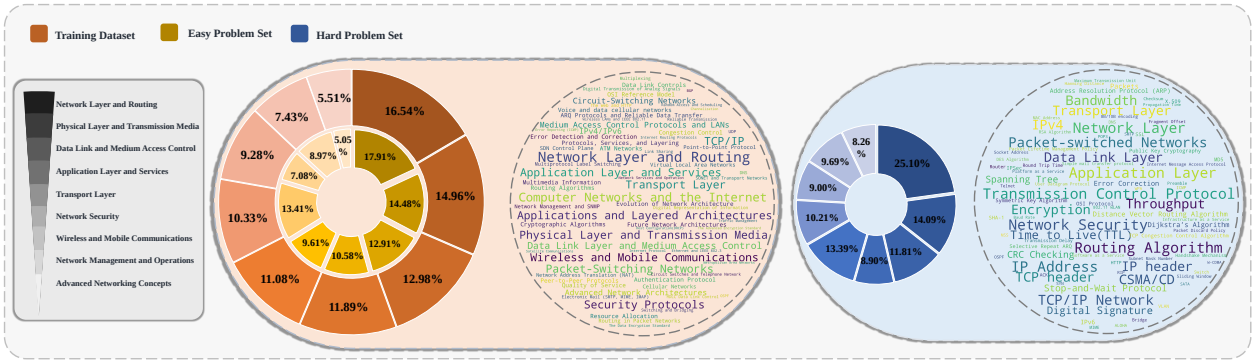

Figure 3. An overview of the released data resource. The three pie charts share the same components listed in the left legend. The legend uses a grayscale graph to indicate each component, with the darker shade corresponding to the deeper-color segments in pie charts (e.g., the darkest orange, yellow, and blue sections match the darkest gray category, which represents the Network layer and Routing).

a router in a network?" or "Which OSI layer is responsible for error detection and correction?". While the 1,718 GPT-generated testing QA pairs can assess a model's grasp of networking concepts, a testing problem set concentrating on logical reasoning and mathematical computation is equally essential for a comprehensive evaluation of the fine-tuned model. For the rest of this paper, we refer to the testing problem set containing 1,718 GPT-generated problems as the "easy problem set" and the 401 manually collected problems as the "hard problem set."

Fig. 3 visually illustrates the data distribution of the training dataset and the easy/hard problem set. In this figure, networking knowledge is categorized into nine distinct sub-domains. The figure shows that, the training data, the easy test problem sets, and the hard test problems, all exhibit a balanced coverage of these sub-domains, thereby ensuring that the model is trained and evaluated with a comprehensive and generalized focus on the networking field. These two word clouds further detail the sub-domain content distribution. The left word cloud highlights the foundational and all-inclusive topics in the training dataset and easy problem set; the right one, on the other hand, reflects the hard problem set focuses on more complex and detailed aspects of networking.

The easy and hard problem sets constitute two critical testing benchmarks that assess an LLM's performance in the networking domain from contrasting perspectives. An additional crucial aspect of the model's evaluation is its performance in a comprehensive and practical problem set with both easy and hard problems. To this end, we randomly down-sampled the easy problem set to create a subset comparable in size to the hard dataset (i.e., 401 QA pairs). Subsequently, we merged this new subset with the hard problem set to form a comprehensive dataset. All three problem sets (easy, hard, and comprehensive) are used in Section V for model evaluation.

V. Experiments

To assess the robustness and generalizability of our proposed method, we conducted a k-fold cross-validation experiment. K-fold cross-validation is a statistical technique used to evaluate machine learning models by partitioning the data into k subsets, using k-1 subsets for training and the remaining subset for testing, and then repeating this process k times with different test subsets. Section IV details the constitution of training data and test data. As we split the GPT-generated QA pairs into training and testing data with ratio 9:1, we opted for a 10-fold cross-validation approach (k = 10) so that folds do not overlap one another. In each iteration, one fold was designated as the test set, while the remaining 9 folds constituted the training data. For each iteration, data augmentation techniques were applied exclusively to the training set, while the corresponding test set remained unaltered. As explained in Section IV-B, this results in approximately 1,718 data for the test set and 370,992 data (15,458 * 24) for the training set in each k-fold iteration. This approach allows us to utilize all available data for both training and testing, providing a more reliable estimate of the model's performance. Fig. 4 presents the accuracies obtained for the folds (i.e., the percentages of correctly answered questions in the test sets).

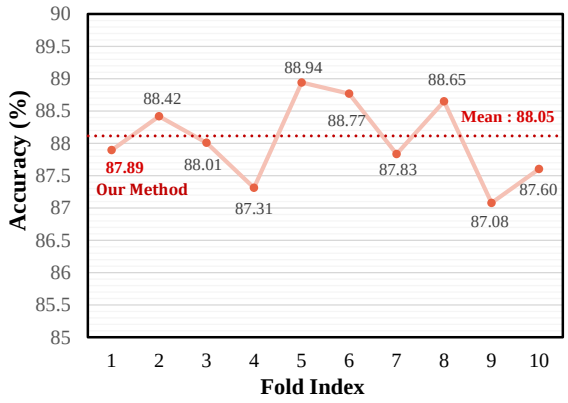

Figure 4. Accuracies of folds 1 to 10 tested on the easy problem.

For the experimental setup, we employed the LLaMA3.1-8B model as our foundational language model and applied LoRA, with rank $r = 8$, employing the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a weight decay of 0. The training process was conducted over 25 epochs with a learning rate of 1e-4, on a four NVIDIA A6000 GPU server.

The results demonstrate consistent performance across all folds, with mean accuracy of 87.89% and minimal variance across folds. This uniformity suggests that our method's efficacy is not contingent upon specific dataset configurations but

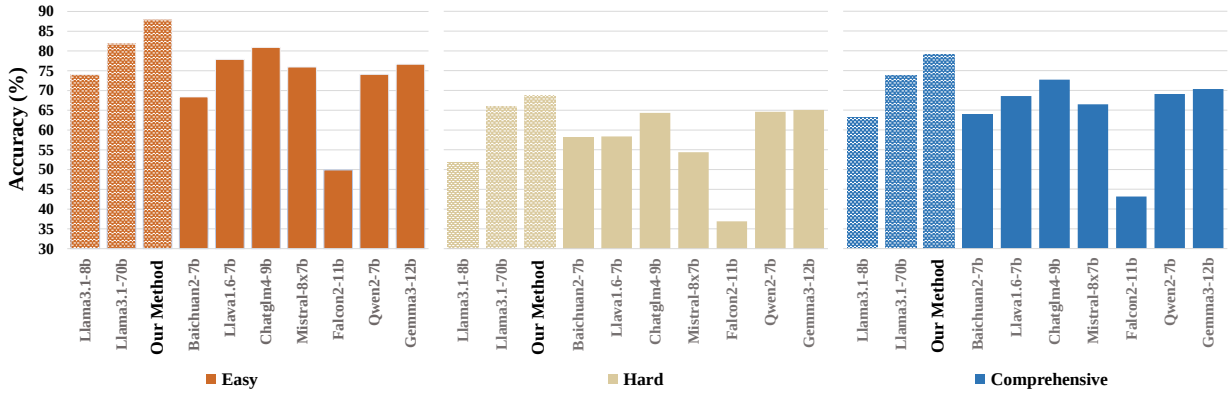

Figure 5. Accuracies of baseline models tested on easy, hard, and comprehensive testing problem sets.

rather represents a generalizable solution. The observed consistency across diverse data subsets underscores the robustness and broad applicability of our proposed approach. Given the consistent performance between each fold, we next select the trained model of fold #1 to compare with other models.

We benchmark our method with different baselines to demonstrate the superiority of our model in the networking domain. Specifically, we compare our method with LLaMA3.1-8B model and LLaMA3.1-70B model, and other representative off-the-shelf LLMs that are 1) released at a similar time as our LLaMA3.1-8B foundational model, and 2) of similar parameter sizes. Fig. 5 presents the accuracy of our model and the above baseline models on the three testing problem sets we built as described in Section IV-B.

For the easy testing problem set, our model correctly answers 88% of questions, significantly surpassing all other baseline models. The original LLaMA3.1-8B model (without fine-tuning) only correctly answers 74% of the questions. The accuracy gap between our fine-tuned model and the foundational LLaMA3.1-8B model indicates the effectiveness of our method when applied to the LLaMA3.1-8B model. Falcon2 11B performs the poorest among all the baseline models with only 50% accuracy. Baichuan2 with accuracy 68% is slightly better than Falcon, but still slightly inferior to the original Llama3.1-8B model. Other baseline models of approximately 8B parameter size, such as ChatGLM4, Qwen2, Mixtral-8x7b, Gemma3, Llava1.6, also demonstrate better performance than the original LLaMA3.1-8B, but fall short of our model, with accuracies ranging from 74% to 80%.

For the hard benchmark that involves complex networking analysis and logical reasoning, our model surpasses all the baseline models with an accuracy of 69%, again demonstrating superior performance than other models tested. Other baseline models with approximately 8B parameters generally achieve accuracies ranging from 54% to 65%. For complex problems, Llama3.1-70B demonstrates the second-best performance among all models tested. This observation aligns well with previously reported observations about the relationship between LLM's reasoning proficiency and model size, i.e., language models with larger sizes are expected to have strong reasoning abilities in general. Notably, our model, despite having about only one-ninth the parameters of LLaMA3.1-70B, still outperforms the larger model. This experimental observation serves as a compelling illustration of the enhanced reasoning capabilities facilitated by the RaC algorithm.

In the comprehensive benchmark, which contains a mix of easy and hard with a ratio of 1:1 (i.e., 401 easy and 401 hard questions), our method maintains its leading position with an accuracy of roughly 79%. The ChatGLM model and Llama3.1-70B follow behind, achieving about 73% accuracy. Other models in this testing benchmark show varying levels of performance, with most falling in the 63-70% accuracy range. Our model's superiority in addressing these multifaceted questions suggests a potential for the model's application in solving real-world problems that require solid knowledge understanding and logical reasoning.

To conclude, we emphasize the effectiveness of our RaC fine-tuning framework by highlighting the following key experimental observations in the three benchmarks:

1) Our model outperforms all baseline models tested. It not only outperforms the largest foundation model of the same family (i.e., the LLaMA3.1 series) with only one-ninth of the parameter size, but it also outperforms all later proposed models with similar sizes.
2) There is an obvious increase in accuracy between our fine-tuned model and the original foundational model (i.e., LLaMA3.1-8B). The accuracy increases from 74% to 87%, 52% to 69%, and 63% to 79% on the easy, hard, and comprehensive benchmarks, respectively.

We conducted ablation studies to evaluate the efficacy of components within our methodology. RaC comprises four key elements: 1) question-answer pairs, 2) problem rephrasing, 3) correct answer explanations, and 4) incorrect answer explanations. Meanwhile, the data augmentation method also significantly influences the LLM's performance. The following experiments gradually remove these components and test the resulting model with the three problem sets introduced above.

We started with the full method using 24x data augmentation. Method A1 excludes incorrect answer explanations from the whole RaC composition; A2 further removes the correct answer explanations from A1; and A3 excludes the problem rephrasing component from A2, leaving only question-answer pairs during fine-tuning. Next, we investigated the influence of data augmentation. Method B1 considers ChoiceBoost, the

weaker 4x data augmentation scheme reported in [12], while B2 completely removes the data augmentation process. Finally, we tested methods A4, A5, and A6, which have the same RaC composition as A1, A2, and A3, but differ in that the training data is augmented 4 times instead of 24 times.

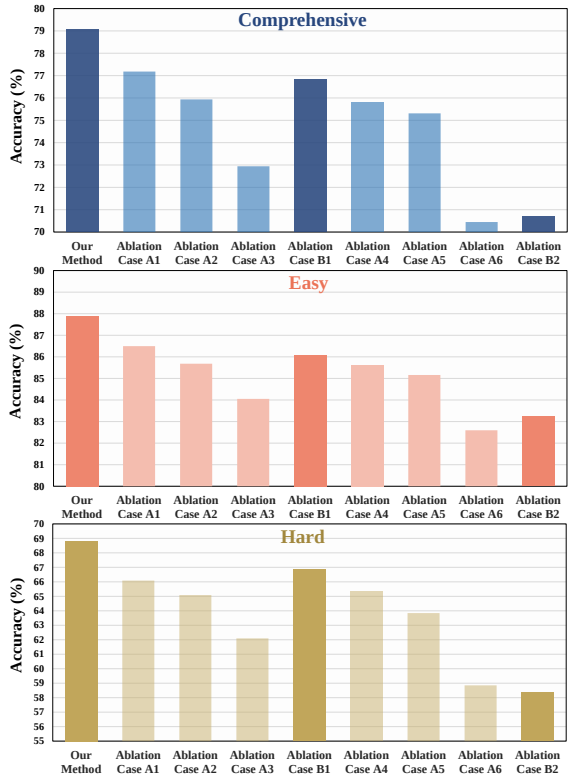

Figure 6. Accuracies of our model, ablation case A1 to A6 and B1 and B2 on three benchmarks.

From the comparison between our method and A1/A2/A3, as well as the comparison between B1 and A4/A5/A6, we see that the sequential elimination of RaC elements from the fine-tuning process led to a progressive decline in performance, underscoring the significance of each RaC component within the framework. Furthermore, from the comparison between our method and B1/B2, we also observe the impressive performance boost brought by the effective 24x data augmentation scheme proposed in this paper. The accuracy gap between our method and B1/B2 shows the advantages of the 24x data augmentation technique over the previously employed 4x data augmentation and underscores the effectiveness of fully eliminating choice ordering biases.

The above experiments also demonstrate the enhanced reasoning ability of the fine-tuned model. Specifically, although the model was fine-tuned using a training dataset containing only simple conceptual knowledge (similar to the easy testing benchmark), it nevertheless achieves improved performance on the hard testing benchmark (see Fig. 5). The hard problem set was manually curated from undergraduate-level textbooks and exam papers, and requires both advanced reasoning skills and complex computations to arrive at the correct answers. As illustrated in Fig. 6(c), both the RaC framework and the data augmentation method contribute significantly to the model's superior performance on these challenging problems.

VI. Conclusion

This paper introduces an efficient LLM fine-tuning framework, referred to as RaC, to address limitations in existing networking LLM: over-reliance on prompting techniques and lack of effective fine-tuning methods for lightweight models. By incorporating question rephrasing and contrastive analysis during fine tuning, RaC significantly enhances LLMs' comprehension and critical thinking abilities. Our experimental results demonstrate a 14.92% accuracy improvement over the foundational model when tested on a comprehensive networking problem set. Further, in the building of the networking dataset used in RaC fine-tuning, we develop a GPT-assisted data mining method for generating high-quality RaC QA pairs and introduce ChoiceBoost-X, a data augmentation technique that expands dataset size while reducing answer bias. For model evaluation, we introduce three new testing benchmarks of varying difficulty, which provide a standardized means of assessing network-oriented LLMs. Our open-source contributions, including the training dataset, three testing benchmarks, the fine-tuned model, and associated codes, facilitate the reproducibility of our work and encourage further advancements in this field. Last but not least, although our investigations center on networking problems, the proposed fine-tuning techniques are generic and applicable to other problem domains.

References

[1] Z. He, A. Gottipati, L. Qiu, X. Luo, K. Xu, Y. Yang, and F. Y. Yan, "Designing network algorithms via large language models," in *ACM HotNets*, 2024, pp. 205–212.

[2] S. K. Mani, Y. Zhou, K. Hsieh, T. Eberl, E. Azulai, I. Frizler, R. Chandra, and S. Kandula, "Enhancing network management using code generated by large language models," in *ACM HotNets*, 2023, pp. 196–204.

[3] R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do llms need to synthesize correct router configurations?" in *ACM HotNets*, 2023, pp. 189–195.

[4] X. Lian, Y. Chen, R. Cheng, J. Huang, P. Thakkar, and T. Xu, "Configuration validation with large language models," *arXiv:2310.09690*, 2023.

[5] H. Cui, Y. Du, Q. Yang, Y. Shao, and S. C. Liew, "LLMind: Orchestrating AI and IoT with LLM for complex task execution," *IEEE Commun. Mag.*, vol. 63, no. 4, pp. 214–220, 2025.

[6] D. Wu, X. Wang, Y. Qiao, Z. Wang, J. Jiang, S. Cui, and F. Wang, "Netllm: Adapting large language models for networking," in *ACM SIGCOMM*, 2024, pp. 661–678.

[7] J. Shao, J. Tong, Q. Wu, W. Guo, Z. Li, Z. Lin, and J. Zhang, "WirelessLLM: Empowering large language models towards wireless intelligence," *arXiv preprint arXiv:2405.17053*, 2024.

[8] M. Kotaru, "Adapting foundation models for operator data analytics," in *ACM HotNets*, 2023, pp. 172–179.

[9] Y. Zhou, N. Yu, and Z. Liu, "Towards interactive research agents for internet incident investigation," in *ACM HotNets*, 2023, pp. 33–40.

[10] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium*, 2024.

[11] J. Zou, M. Zhou, T. Li, S. Han, and D. Zhang, "Promptintern: Saving inference costs by internalizing recurrent prompt during large language model fine-tuning," *arXiv:2407.02211*, 2024.

[12] L. Wang, Y. Du, J. Lin, K. Chen, and S. C. Liew, "Rephrase and contrast: Fine-tuning language models for enhanced understanding of communication and computer networks," *IEEE ICNC*, 2025.

[13] R. Nikbakht, M. Benzaghta, and G. Geraci, "TSpec-LLM: An open-source dataset for llm understanding of 3GPP specifications," *arXiv:2406.01768*, 2024.

[14] A. Maatouk, F. Ayed, N. Piovesan, A. De Domenico, M. Debbah, and Z.-Q. Luo, "Teleqna: A benchmark dataset to assess large language models telecommunications knowledge," *arXiv:2310.15051*, 2023.

[15] Y. Lin, R. Zhang, W. Huang, K. Wang, Z. Ding, D. K. So, and D. Niyato, "Empowering large language models in wireless communication: A novel dataset and fine-tuning framework," *arXiv:2501.09631*, 2025.