Rephrase and Contrast: Fine-Tuning Language Models for Enhanced Understanding of Communication and Computer Networks

Liujianfu Wang[a†], Yuyang Du[a†], Jingqi Lin[b†], Kexin Chen[a], Soung Chang Liew[a*]
[a] The Chinese University of Hong Kong, Hong Kong SAR, China
[b] Huazhong University of Science and Technology, Wuhan, China

***Abstract*—Large language models (LLMs) are being widely researched across various disciplines, with significant recent efforts focusing on adapting LLMs for understanding of how communication networks operate. However, over-reliance on prompting techniques hinders the full 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 our Rephrase and Contrast (RaC) framework, an efficient fine-tuning framework. RaC enhances LLMs' comprehension and critical thinking abilities by incorporating question reformulation and contrastive analysis of correct and incorrect answers during the fine-tuning process. Experimental results demonstrate a 63.73% accuracy improvement over the foundational model when tested on a comprehensive networking problem set. Moreover, to efficiently construct the dataset for RaC fine-tuning, we develop a GPT-assisted data mining method for generating high-quality question-answer (QA) pairs; furthermore, we introduce ChoiceBoost, a data augmentation technique that expands dataset size while reducing answer-order bias. Apart from these technical innovations, we contribute to the networking community by open-sourcing valuable research resources, including: 1) the fine-tuned networking model referred to as RaC-Net, 2) the training dataset used for fine-tuning the model, 3) three testing problem sets of different difficulties to serve as benchmarks for future research, and 4) code associated with the above resources.**

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

I. Introduction

In recent years, artificial intelligence has witnessed a paradigm shift with the advent of large language models (LLMs) that exhibit near-human intelligence across various domains [1]. These models have captured the attention of researchers and practitioners from diverse fields. The remarkable capabilities of LLMs have sparked a surge of interest in their potential applications across various scientific disciplines.

The field of communication networks is no exception to this trend. Recent studies have explored the application of LLMs in several key areas within the networking domain. For instance, researchers have investigated the use of LLMs in network topology analysis [2], [3], wireless device configuration [4], [5], network setup optimization [6]–[8], network diagnosis [9], [10], and network security [11]. These efforts highlight the potential of LLMs to enhance the performance and efficiency of various network-related tasks.

Despite the considerable efforts, current approaches to integrating LLMs in communication networks suffer from several drawbacks. One drawback is the use of prompt tuning techniques. While prompt tuning methods such as chain of thought (CoT) enable rapid proof-of-concept demonstrations, they are inherently inefficient and often clumsy. Crafting effective prompts may require extensive trial-and-error efforts, and the resulting solutions may lack robustness and generalizability because the foundational parameters within LLMs are not modified to tailor for networking tasks. Moreover, the computational overhead associated with prompt tuning can be substantial, especially for large-scale network applications [12].

Fine-tuning methods that adjust the parameters of LLMs for networking tasks have the potential to overcome such drawbacks of prompt engineering. One technique involves using question-answer (QA) pairs to refine the model's parameter weights to improve its responses to questions [13]. This paper introduces an efficient LLM fine-tuning scheme referred to as the Rephrase and Contrast (RaC) framework. Unlike conventional fine-tuning methods that only use question-answer (QA) pairs, RaC supplies the LLM with additional information, including 1) a reformulation of the question asked, and 2) a contrastive analysis of both the correct answer and potential incorrect answers. This approach, which effectively mimics how humans learn, enhances problem comprehension by LLMs and significantly improves their critical thinking ability essential for analyzing complex networking issues. Experimental results on RaC indicate a 63.73% accuracy improvement compared to the foundational model without fine-tuning when tested on a comprehensive networking problem set.

We remark that this paper only focuses on endowing LLMs with foundational networking knowledge. This is an essential first step before we can further train LLMs to perform specific networking tasks such as network configuration, network management, resource allocation, security management, and autonomous device coordination in response to environmental changes and human directives. A crucial cornerstone of achieving fully AI-automated network management and control is ensuring that LLMs understand how networks operate. In this context, our investigation serves as a critical intermediate step

The work was supported in part by the 2024 Shenzhen-Hong Kong-Macao Science and Technology Program (STIC) Category C Project, under Grant No. SGDX20230821094359004.

Code, data, and model availability: https://github.com/1155157110/RaC.

†L. Wang, Y. Du, J. Lin contribute equally. This work was completed during J. Lin's internship at the Chinese University of Hong Kong.

*S. C. Liew (soung@ie.cuhk.edu.hk) is the corresponding author.

toward these ultimate goals.

Apart from the RaC fine-tuning framework, we notice a lack of datasets for training LLMs on networking tasks. To address this, we develop a GPT-assisted data mining method and introduce ChoiceBoost, a simple yet highly effective data augmentation technique for building training datasets.

For data mining, we leverage the language processing ability of GPT-4 to analyze raw materials extracted from classic networking textbooks and generate high-quality QA pairs that meet the needs of RaC fine-tuning. The QA pair generation process applies the in-context learning (ICL) technique to guide the GPT-4 model in structuring the data and help the model better understand what output is desired in each step.

For data augmentation, ChoiceBoost augments the dataset by altering the order of the correct answer and incorrect answers in multiple-choice questions. This approach expands the dataset size by four times, maintaining the same knowledge content while offering varied choice arrangements. ChoiceBoost effectively reduces bias related to answer order and reinforces the model's understanding of the material. Subsequent ablation studies show a clear improvement in the model's performance attributed to ChoiceBoost.

Beyond our technical contributions (i.e., RaC, the GPT-assisted data mining scheme, and ChoiceBoost), we are also releasing several valuable resources to the networking community. These include: 1) our training dataset used for LLM fine-tuning, 2) three novel testing problem sets that can serve as testing benchmarks for future research, 3) the fine-tuned models referred to as the RaC-Net, and 4) all code associated with the above three components. This comprehensive release will enable researchers and practitioners to reproduce our results and build upon our work to develop advanced network-oriented LLMs.

II. Related Work

Recent advances in LLMs have inspired many researchers to explore their adaptation for networking tasks [2]–[11], [14], [15]. 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 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 [8], 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 [2], [4]–[6], [8]–[11], few-shot prompting methods, such as in-context learning (ICL), are used to guide LLMs to generate answers reliably. Chain of thought (CoT) was applied in [2], [6], [8], [10] to enhance LLMs' reasoning ability. Additionally, retrieval methods like retrieval augmented generation (RAG) are used in [8], [14] 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 adapted LLMs through fine-tuning for wireless networking tasks, [7] adapted a LLaMA-2 7B 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, [7] did not make new contributions in fine-tuning methods: it directly 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 [3], [5], [14], and [15] released their research data. However, the released datasets are either too small for LLM adaptations [3], [5], 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 [14], [15]. Compared with previous research, we provide a dataset that is self-contained as far as LLM adaptation is concerned. Additionally, we also provide the community with the source code associated with the model fine-tuning, facilitating reproductions and follow-ups of this work.

In addition, recognizing the absence of a generally accepted testing benchmark for evaluating the performance of LLMs 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

This section introduces RaC, a framework designed to enhance the efficiency of model fine-tuning. RaC is inspired by two observations:

Observation 1: The Significance 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 the detailed context of the question deepens the understanding by the responder to understand.

Observation 2: Wrong Answer Matters. In human learning, analyzing incorrect answers enhances understanding. By contrasting them with correct answers, we gain a deeper understanding of the topic, fostering critical thinking. Additionally, explanations of wrong answers can introduce new information not covered in the correct answer's explanation.

Based 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 provides an example to illustrate the process.

Before delving into Fig. 1, let us briefly explain why we use the multiple-choice format as the target QA format in RaC-Net. This format offers an objective and easy-to-grade method

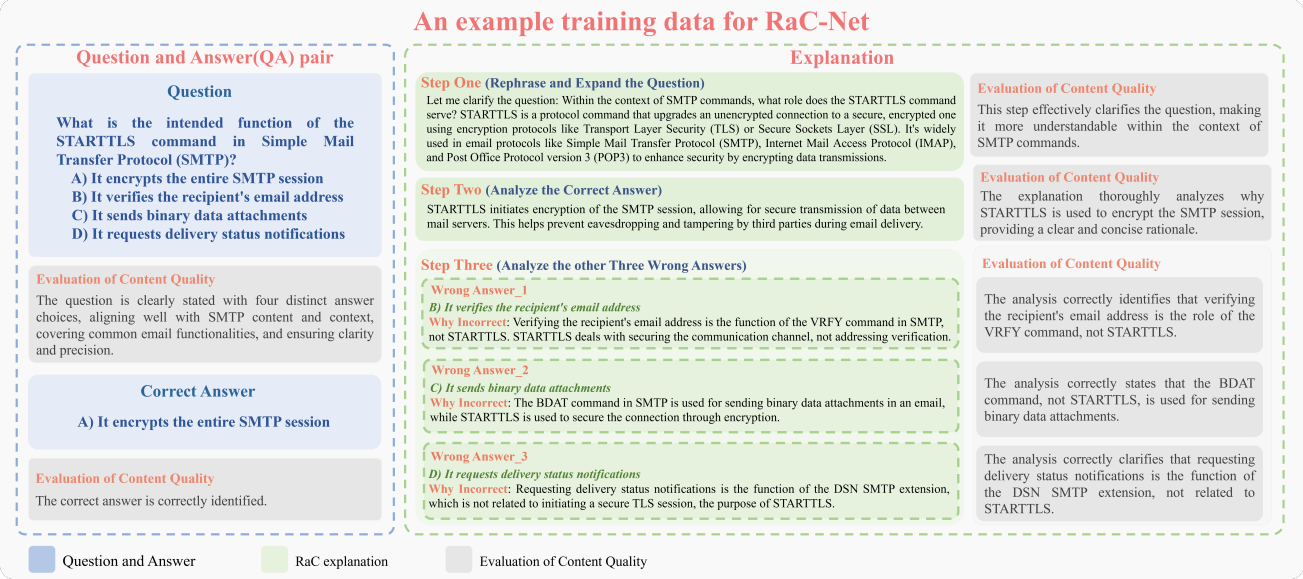

Figure 1. An example of training data that meets the data structure requirements of our RaC framework. For easier understanding and better assessment of data quality, comments have been added in the grey box. These comments are not part of the training dataset; they are included for illustrative purposes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

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 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 worth noting that all of the question reformulation contents and contrastive analysis are autonomously generated by the GPT model, without human intervention.

For the RaC implementation, our framework can be adopted in a variety of supervised fine-tuning methods. We implement LoRA fine-tuning 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 highlight that the QA pair we are looking for is more than a simple combination of a question and an answer. A core feature of our QA pair 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, the rotation-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 10 textbooks on computer networking, which include both foundational theories and the latest technical advancements in the networking field. Each textbook was chosen for its unique technical insights, thus ensuring comprehensive coverage of the networking field.

In our dataset, each instance is formatted as 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 well 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) software 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. We segmented the textual content by section to accommodate GPT-4's token limit without harming the context coherence within each segment. 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 the implementation of QA pair generation, 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 this QA pair generation process. The process begins by describing the basic task for the model, which specifies 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.

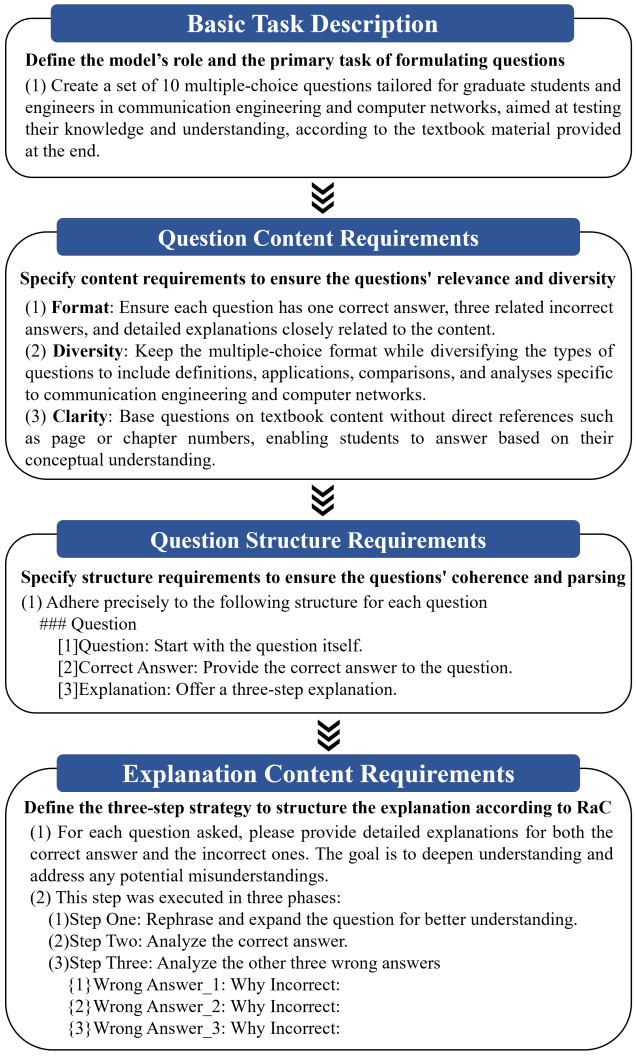

Figure 2. LLM prompt template used by RaC-Net for creating QA pairs.

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.

After the GPT-assisted QA pair generation, we now explain how we augment the obtained dataset with ChoiceBoost. 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.

ChoiceBoost is designed to mitigate the above two issues. In ChoiceBoost, each question in the dataset was duplicated four times, with each duplicate assigned a different correct answer (A, B, C, or D), while maintaining the original order of incorrect options. This approach substantially expands the dataset without additional API cost (which enhances the model's capacity by learning from diverse answer configurations) and mitigates biases inherent in the initial data generation process.

B. Data Resources Released

The initial dataset derived from ten textbooks without post-processing data augmentation comprises 15,119 QA pairs. These original QA pairs encompass essential sub-domains knowledge of computer networking described in each individual textbook, such as layered network architecture, protocols design, network security, and network management. From the raw data, 756 QA pairs (approximately 5%) were randomly selected to form a testing problem set. We then used ChoiceBoost to augment the remaining 15,119 – 756 = 14,363 QA pairs, resulting in an unbiased training dataset containing 14,363 * 4 = 57,452 QA pairs.

In addition to the GPT-generated QA pairs, we manually collected 327 multiple-choice testing questions closely aligned with the content of the ten 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 a router in a network?' or 'Which OSI layer is responsible for error detection and correction?'. While the 756 GPT-generated testing QA pairs can assess a model's grasp of networking concepts, a testing problem

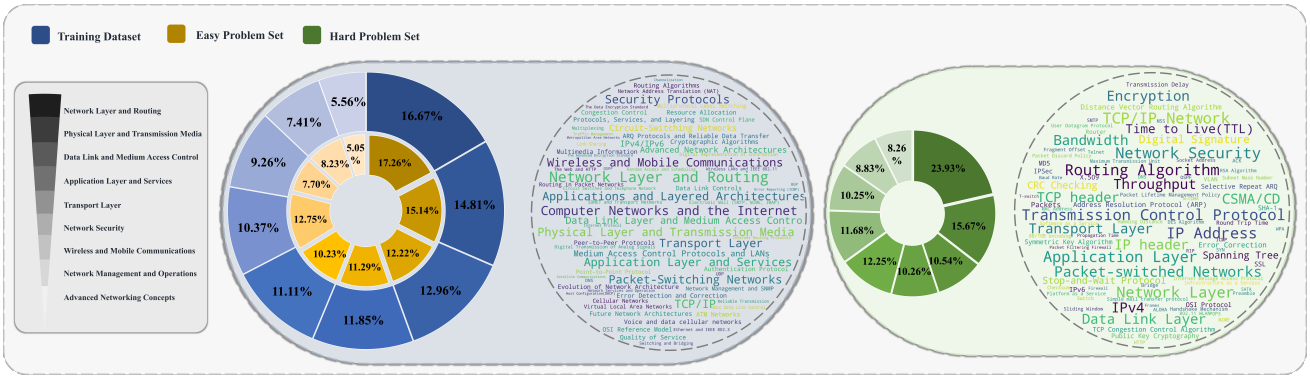

Figure 3. An overview of the released data resource. The three pie charts share the same components listed in the left legend. The legend uses a grayscale graph to indicate each component, with the darker shade corresponding to the deeper-color segments in pie charts (e.g., the darkest blue, orange,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

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 756 GPT-generated problems as the "easy problem set" and the 327 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., 327 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. In section IV we detail the constitution of training data and test data. As we split the GPT-generated QA pair into training data and testing problem set with a ratio of 19:1, we opted for a 20-fold cross-validation approach (k = 20) to ensure comprehensive evaluations, where folds do not overlap with one another. In each iteration, one fold was designated as the test set, while the remaining 19 folds constituted the training data, to which data augmentation techniques were applied before model training. With respect to the explanation given in Section IV.B, this methodology results in approximately 756 data for the test set and 57,452 data (14,363 * 4) for the training set in each k-fold iteration. This approach allows us to utilize all available data for both training and testing, thus providing a more reliable estimate of the model's performance. For each iteration, data augmentation techniques were applied exclusively to the training set, while the corresponding test set remained unaltered to maintain the integrity of the evaluation process. Fig. 4 illustrates the accuracy metrics obtained for each fold.

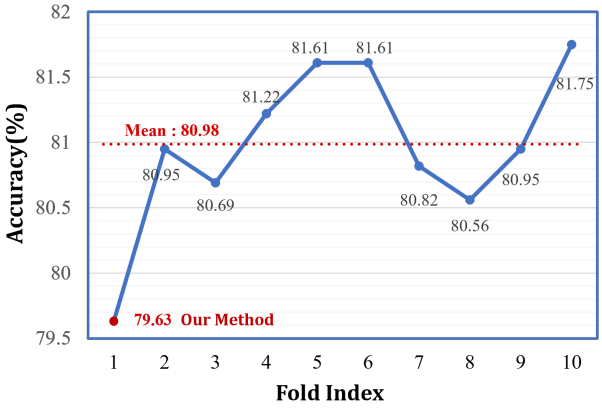

Figure 4. Accuracies of models of fold index 1 to 10 tested on the easy problem set. Due to the limitation in our computational resources, we randomly selected 10 folds among the 20 candidate folds for concept proofing.

For the experimental setup, we employed the LLaMA-2-7B model as our foundational language model and applied LoRA fine-tuning. The training process was conducted over 35 epochs with a learning rate of 1e-4. Computational resources consisted of four NVIDIA A6000 GPU cards to facilitate efficient parallel processing.

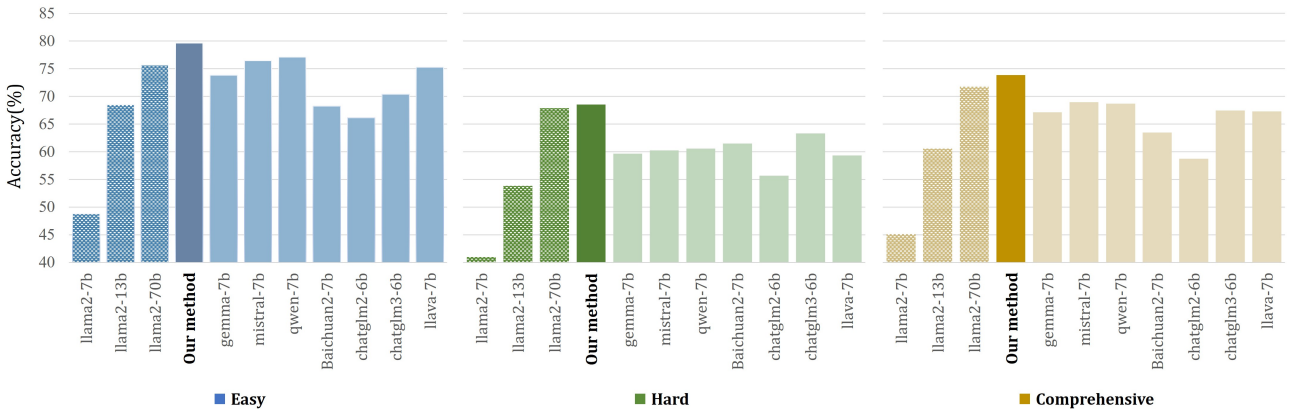

Figure 5. Accuracies of baseline models tested on easy, hard, and comprehensive testing problem sets.

The results demonstrate a consistent performance across all folds, with a mean accuracy of 80.98% accuracy across different folds and minimal variance in accuracy scores. This uniformity suggests that our method's efficacy is not contingent upon specific dataset configurations but 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 here select the trained model of fold #1 as our method 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 LLaMA-2 models of various sizes (i.e., 7B, 13B, and 70B) and representative off-the-shelf LLMs that are 1) released at a similar time as our LLaMA-2 7B foundational model does, and 2) of similar parameter sizes. Fig. 5 presents the accuracy of our model and all the above baseline models on the three testing problem sets we built in Section IV-B.

For evaluation results on the easy testing problem set, our model correctly answers 80% of questions, significantly surpassing all other baseline models in terms of accuracy. The original LLaMA-2 7B model (without fine-tuning) performs the poorest among all tested models, with only 48% accuracy on the tested dataset. The huge accuracy gap between our fine-tuned model and the foundational LLaMA-2 7B indicates the effectiveness of our method when applied to the LLaMA-2 7B model. Other baseline models of approximately 7B parameter size, such as ChatGLM and BaiChuan, also demonstrate better performance than LLaMA-2 7B, but still fall short of our model, with accuracies ranging from 66% to 70%.

For the hard testing 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 7B models show a significant decrease in accuracy (compared to the easy testing benchmark) due to the increased problem difficulty, with accuracies generally falling between 55% and 63%. For complex problems, LLaMA-2 70B demonstrates the second-best performance among all models tested and shows a substantial lead over the third-best model (i.e., ChatGLM3-6B). This observation aligns well with previously reported observations [16] 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. Our model, despite having only one-tenth the parameters of LLaMA-2 70B, still marginally 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 tasks, our method maintains its leading position with an accuracy of roughly 75%. The LLaMA-2 70B model follows closely behind, achieving about 72% accuracy. Other models in this testing benchmark show varying levels of performance, with most falling in the 65-70% accuracy range. Our model's superiority in effectively addressing these multifaceted questions suggests a potential for the model's application in solving real-world problems, which require both 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 testing benchmarks:

1) Our model outperforms all baseline models tested. It not only outperforms the largest foundation model of the same family (i.e., the LLaMA-2 series) with only one-tenth 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., LLaMA-2 7B). The accuracy increases from 47% to 80%, 41% to 68%, and 45% to 74% on the easy benchmark, hard benchmark, and comprehensive dataset, respectively.

To assess the efficacy of individual components within our methodology, we conducted a series of ablation studies. The Rephrase and Compare (RaC) framework comprises four key elements: (1) question-answer pairs, (2) problem rephrasing, (3) correct answer explanations, and (4) incorrect answer explana-

tions. Our ablation experiments involve systematic removals of these components, followed by an evaluation of the resultant model's performance on a comprehensive testing benchmark. Specifically, we implemented three ablation cases: A1, which eliminated incorrect answer explanations; A2, which further removed correct answer explanations from A1; and A3, which excluded the problem rephrasing component from A2, leaving only question-answer pairs in the fine-tuning process. As illustrated in Fig. 6, the sequential elimination of these elements from the RaC framework led to a progressive decline in performance, underscoring the significance of each component within the framework.

Additionally, we examined an alternative ablation case (designated as case B) that omitted the data augmentation method delineated in Section IV. A comparative analysis between our proposed method and ablation case B revealed a substantial reduction in the model's answer accuracy. This empirical observation emphasizes the critical role of our data augmentation scheme in enhancing overall performance.

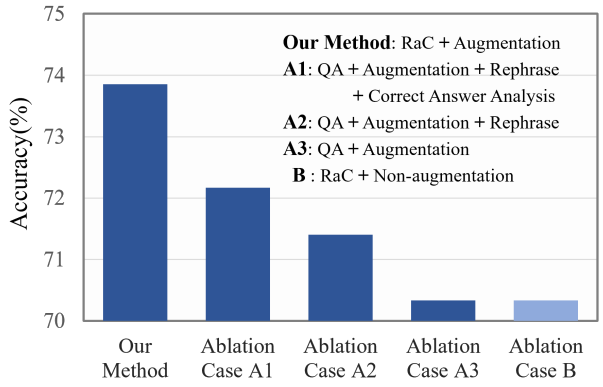

Figure 6. Accuracies of our model, ablation case A1 (our model - wrong answer explanation), A2 (A1 - correct answer explanation), A3 (A2 - question rephrasing) on the comprehensive dataset.

VI. Conclusion

This paper introduces RaC, an efficient fine-tuning framework for LLM. RaC addresses the limitations arising from over-reliance on prompting techniques and the lack of effective fine-tuning methods for lightweight models. By incorporating question reformulation and contrastive analysis during fine-tuning, RaC significantly enhances LLMs' comprehension and critical thinking abilities within networking contexts. Our experimental results demonstrate a 63.73% accuracy improvement over the foundational model when tested on a comprehensive networking problem set. Furthermore, to obtain the dataset used in RaC fine-tuning, we develop a GPT-assisted data mining method for generating high-quality RaC QA pairs and introduce ChoiceBoost, 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.

This work marks a significant step towards more effective LLMs for networking applications, bridging the gap between general-purpose language models and domain-specific networking knowledge. Future work could explore the scalability of RaC to larger models and diverse networking tasks, investigate transfer learning between different networking domains, develop more sophisticated data augmentation techniques, and conduct user studies in real-world scenarios.

VII. Acknowledgment

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

References

[1] Q. Niu, J. Liu, Z. Bi, P. Feng, B. Peng, and K. Chen, "Large language models and cognitive science: A comprehensive review of similarities, differences, and challenges," *arXiv preprint arXiv:2409.02387*, 2024.

[2] Z. He, A. Gottipati, L. Qiu, F. Y. Yan, X. Luo, K. Xu, and Y. Yang, "Llm-abr: Designing adaptive bitrate algorithms via large language models," *arXiv preprint arXiv:2404.01617*, 2024.

[3] S. K. Mani, Y. Zhou, K. Hsieh, S. Segarra, T. Eberl, E. Azulai, I. Frizler, R. Chandra, and S. Kandula, "Enhancing network management using code generated by large language models," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 196–204.

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

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

[6] H. Cui, Y. Du, Q. Yang, Y. Shao, and S. C. Liew, "Llmind: Orchestrating ai and iot with llms for complex task execution," *arXiv preprint arXiv:2312.09007*, 2023.

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

[8] 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.

[9] M. Kotaru, "Adapting foundation models for operator data analytics," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 172–179.

[10] Y. Zhou, N. Yu, and Z. Liu, "Towards interactive research agents for internet incident investigation," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 33–40.

[11] 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 (NDSS)*, 2024.

[12] 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 preprint arXiv:2407.02211*, 2024.

[13] K. Chen, Y. Du, T. You, M. Islam, Z. Guo, Y. Jin, G. Chen, and P.-A. Heng, "LLM-assisted multi-teacher continual learning for visual question answering in robotic surgery," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[14] R. Nikbakht, M. Benzaghta, and G. Geraci, "Tspec-llm: An open-source dataset for llm understanding of 3gpp specifications," *arXiv preprint arXiv:2406.01768*, 2024.

[15] 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 preprint arXiv:2310.15051*, 2023.

[16] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.