2025 IEEE 26th International Conference on High Performance Switching and Routing (HPSR) | 979-8-3315-2991-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/HPSR64165.2025.11038889

Partitioning Prompts for Higher Efficacy in Network Design with Large Language Model

Vishnu Komanduri, Scott Alessio, Sebastian Estropia, Gokhan Yerdelen, Tyler Ferreira, Murali Gunti, Ziqian Dong, and Roberto Rojas-Cessa

***Abstract*—In this paper, we propose deliverable partitioning in prompt design to assist Large Language Models (LLMs) in improving response correctness for network design and configuration. While recent research has explored the use of LLMs to enhance network management efficiency, their responses often remain inconsistent, incomplete, or inaccurate. Often, LLM-generated configurations contain missing or erroneous configuration commands, which can lead to operational failures. Our proposed partitioning methodology aims to mitigate these issues by decomposing complex network configuration tasks into simplified and focused tasks. To evaluate the effectiveness of this approach, we introduce a scoring policy and conduct extensive experiments across three levels of network complexity and varying degrees of design choice ambiguity. We also compare the performance of leading LLMs, including ChatGPT, Copilot, and DeepSeek. Our findings indicate that partitioning the inquiry process leads to more accurate and consistent responses than non-partitioned approaches, especially in scenarios where design parameters are explicitly defined and leave some but small room, as ambiguity, for inference.**

I. Introduction

Large Language Models (LLMs), as a representation of Natural Language Processing, boast tremendous capabilities for solving a wide variety of tasks that require knowledge and expertise across numerous disciplines. LLMs make this possible by being trained on an impressive sea of information [1]. Existing LLMs are primarily trained for general-purpose tasks such as drafting personal and professional documents or summarizing passages. Yet, they have also demonstrated knowledge of specialized topics, including engineering and design. Their remarkable ability to interpret and generate human language, as well as identify patterns and structures within large amounts of information, makes them powerful tools for supporting a variety of personal and professional tasks. The rapid and widespread adoption of LLMs for generating and extracting information [2] offers a glimpse into how integrating artificial intelligence (AI) with human capabilities can enhance productivity across a wide range of activities.

In human-LLM interactions, the LLM analyzes the user's input prompt to identify intent and scours its vast knowledge to generate a satisfactory response. However, due to the broad range of information used during training, general-purpose LLMs may not yet be fully equipped for specialized domains. Although they demonstrate impressive abilities in basic reasoning and logical tasks, LLMs often struggle to apply critical thinking and in-depth analysis to complex engineering problems that require domain-specific expertise.

Network engineering tasks—such as designing, provisioning, modifying, or removing network elements—are often arbitrary and vary significantly in complexity and purpose. While specialized LLMs may be trained to handle such diverse tasks, general-purpose LLMs may fall short or produce error-prone solutions. Therefore, it is essential to explore strategies that enhance their network configuration capabilities. One such approach involves prompt engineering—crafting simplified, focused input queries to reduce ambiguity and improve user-LLM communication. Experiments have shown that LLMs can struggle with complex prompts, often resulting in inaccurate responses. Verbose or unclear prompts can obscure the user's intent and derail the model's output. Additionally, long and intricate inputs may exceed the LLM's memory limits, potentially degrading the quality of the response. These findings suggest that prompting plays a critical role in effective LLM interaction. While some strategies propose using coded schematics for network configuration prompts [3], we argue that well-designed text prompts may be sufficient to achieve optimal results.

To improve the correctness of LLM responses, we investigate the following hypothesis: a partitioned prompting process increases the likelihood of generating LLM responses with near-optimal accuracy for network configuration tasks. To test this hypothesis, we first introduce scoring policies to evaluate the accuracy of LLM responses to identify errors. We then propose partitioning the query process to enhance the clarity and correctness of prompts related to network configuration. The underlying intuition is that process partitioning evidences the user's intent and simplifies the structure of the request, making it easier for the LLM to interpret. This strategy can also be combined with the partitioning of network elements to further streamline the prompt. We conduct extensive evaluations of the proposed approach across leading LLMs and assess the results using our scoring policy. Additionally, we design the experiments to account for prompt ambiguity and varying levels of network complexity, allowing us to categorize prompt difficulty. Our results demonstrate that the proposed approach achieves a high level of correctness, as measured by our scoring policy. This result marks additional improvement over prior work using schematic prompts for network design and description [3].

The remainder of the paper is organized as follows. Section II discusses related work on LLMs used for network

V. Komanduri, S. Alessio, S. Estropia, G. Yerdelen, T. Ferreira, Murali Gunti, and R. Rojas-Cessa are with New Jersey Institute of Technology, Newark, NJ 07102. Z. Dong is with New York Institute of Technology, New York, NY 10023. Email: vk379@njit.edu

configuration. Section III discusses the proposed process-prompt partition method and the scoring policy created for the evaluation of the LLM response, and complexity and ambiguity as factors that affect response correctness. Section IV presents the test set up and results of our evaluations. Section V presents our conclusions.

II. Related Work

The use of LLMs for network configuration and design is becoming of large interest [4]. Network implementation is often complex, where even minor methodological intricacies can significantly impact the overall system performance [5]. Applying verification technologies in the networking domain has proven challenging, as most state-of-the-art approaches rely on rule-based methods or direct human intervention [6]. Rule-based verification is tedious and requires a vast set of rules to accommodate the wide range of potential model responses. Meanwhile, human intervention, often seen as the ultimate solution to LLM verification, undermines the goal of full automation. Additionally, because LLMs primarily use text as their medium of expression, their responses are inherently subjective, making verification ambiguous. To address these challenges, recent work has proposed innovative verification methods by converting text-based responses into alternative, more analyzable representations. Besta et al [7] and Sun et al. [8] proposed a graph-based approach that encodes LLM responses visually, enabling enhanced analysis and clarity. Visualizing LLM outputs through alternative mediums reduces textual ambiguity and enhances specificity.

Although response verification is crucial for autonomous network implementation systems, it does not eliminate the possibility of generating invalid, incorrect, or incomplete responses [9]. This challenge arises primarily from two factors: the absence of networking-specialized LLMs and the inherent randomness of general-purpose models, which can occasionally produce hallucinations—responses that are either irrelevant to the input or partially formed [10]. As a result, zero-touch network and service management systems often rely on supplementary techniques such as prompt engineering and domain-specific training [11]. Other communication beyond text has been also tested, using schematics on text-based LLM inputs [3]. It is reported that LLMs detection of network topology improves but the responses from LLMs continue to inject errors.

One promising solution is NetLLM, which enhances general-purpose LLMs by incorporating network-specific knowledge, allowing them to adapt to a broad range of networking scenarios [12]. This strategy improves model performance without requiring the development of entirely new systems. In contrast, Ifland et al. [13] introduced a networking-focused LLM built from the ground up using OpenAI's ChatGPT as a foundation.

III. Input-Prompt Enhancements

Enhancing an input prompt can reduce ambiguities in complex problem statements and simplify the context for better comprehension and analysis of the objective. While traditional prompt-enhancement methodologies may lead to modest improvements in response accuracy, determining the optimal amount of information to include remains a challenge. LLM responses often degrade when provided with either too little or too much information, mirroring how the human brain may struggle with tasks that involve unbalanced information loads.

We hypothesize that a concise yet well-specified input is more effective. Although this approach is conceptually intuitive, its practical application is challenging, especially when condensing complex network descriptions. Graph-based schematic prompts offer one way to represent network topologies for LLM interpretation [3], but they come with their own limitations. In this work, we explore how to design effective text-based prompts as a general input method for LLMs.

A. Partitioning

We propose a prompt-partitioning approach, which breaks down a large, information-dense prompt into smaller, more manageable segments that a language model can process more effectively. We divide this partitioning process into network partitioning (information given to the LLM) and process partitioning (information requested to the LLM). Therefore, partitioning can be applied either to the network description or to the inquiry process—that is, the requested response. In this work, we propose the latter. The process involves dissecting a verbose prompt into a series of smaller task descriptions, submitting each task to the LLM sequentially, evaluating their individual responses, and finally aggregating the overall response accuracy once all tasks have been addressed, as illustrated in Figure 1.

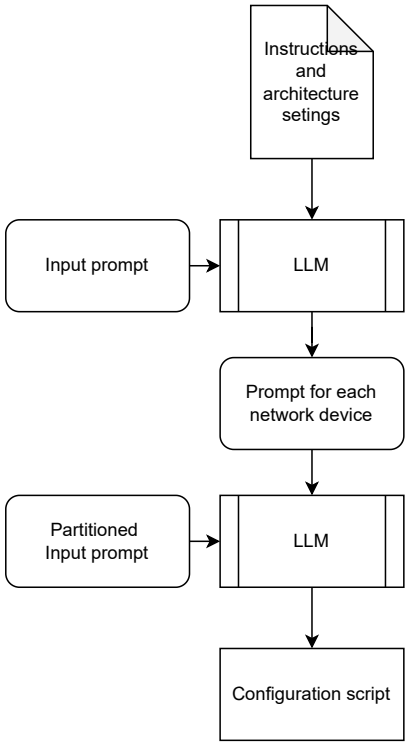


Fig. 1: Example of network partitioning for a single task.

a) Network Partitioning: Because partitioning can be applied in various ways, it is important to evaluate different levels of granularity to analyze trends in response accuracy. We theorize that, ideally, a partitioned prompt for network configuration would involve only a single network element. However, the optimal level of partitioning may vary depending on the network's complexity and the capabilities of the LLM being used. Table I presents different partitioning levels, organizing them from the most granular (i.e., partitioning by individual network elements) to the least. Each level corresponds to a different number of prompts, with the most granular level involving the highest number of individual prompts. Once the network has been partitioned, a consolidation prompt must be crafted to combine all previous responses and generate a comprehensive output for further evaluation.

TABLE I: Network partitioning levels.

Partitioning	Level (%)
Partitions each network element separately	100
Partitions network into unique subnets	75
Partitions based on the type of network element	50
Separates network and summary prompt	25
Contains all information	0

b) Deliverable Partitioning: Different from network partitioning, in deliverable partitioning the intent or requested deliverable is singled out in a prompt. Moreover, the network is fully described at once, therefore providing all network information to the LLM. Therefore, the requested task is a single operation in what the LLM can focus in. The process is then simpler than in network partitioning.

B. Response Scoring Policy

Although partitioning addresses the challenge of improving response correctness, proper frameworks to grade a received response need to be established for verification. As we delve into the evaluation of both network, as well as deliverable partitioning, we define unique grading schemes for each purpose.

a) Scoring policy for Network Partitioning: Network partitioning aims to reduce the number of network elements an LLM needs address in order to provide an optimal response, and because networks are intrinsically complex, it is difficult to potentially identify the various types of errors an LLM could generate. For this reason, we categorize the scoring policy of such partitioned responses into four types of errors: *Incorrect command*, *Missing commands*, *IP addressing errors*, and *Topology errors* and allocate 25 penalty points for each type. An ideal solution without any error would be scored 100 points. Each high-level error is further fine-tuned into smaller error in that category with partial score deductions.

Table II shows the proposed scoring methodology that is performed at the end of the final prompt. Unlike traditional pass/fail systems, our proposed scoring system provides insights on the effectiveness of LLM responses. The culmination of a prompt-enhancement methodology, and response-scoring framework, allow for proper evaluation of an LLM in a wide range of network configuration scenarios.

b) Scoring Policy for Deliverable Partitioning: When evaluating deliverable partitioning, it is essential to assess each deliverable independently, ensuring that the LLM is guided by a single, well-defined objective at a time. Unlike prompt-based partitioning, deliverable partitioning requires a more granular evaluation approach due to the increased potential for variability across multiple outputs. To address this, we introduce an alternative scoring system specifically tailored to capture the nuances of multi-deliverable responses. This system fairly accounts for variability and includes additional error metrics to provide a more comprehensive assessment of LLM performance.

Table III outlines the proposed scoring methodology, which uniformly penalizes errors in LLM-generated responses while maintaining a balanced approach between functions that LLMs perform well and those that do not. In the context of deliverable partitioning, this methodology aims at providing a fair treatment of all error types by assigning consistent weight across different categories, thereby supporting an objective and comprehensive evaluation.

TABLE II: Proposed scoring scheme for network partitioning.

Type of Error	**Penalty (%)**
Incorrect Command	**25**
Invalid Syntax	12.5
Configure Terminal omitted	6.25
Spelling Mistake	6.25
Missing Commands	**25**
No Routing Commands provided	12.5
Not all interfaces are configured	12.5
IP Addressing Errors	**25**
Wrong IP incl. the first octet	12.5
Wrong IP address excl. first octet	12.5
Incorrect subnet mask	6.25
Topology Errors	**25**
Wrong number of network elements	12.5
Invalid Links	10
Incorrect edge router configuration	2.5

IV. Evaluation

We test our prompts on ChatGPT-4o, Microsoft Copilot, and Deepseek, across network topologies of varying complexity to identify observable trends. We apply partitioning of networks and processes to identify the most effective approach and to assess the strengths and weaknesses of each approach in the context of network configuration. For each scenario, responses are evaluated using the proposed scoring policy.

A. Experiment 1: The Impact of Network Partitioning

For clarity, we initially employ prompts with minimal ambiguity but with all the necessary features of network configuration, such as IP addresses, subnet masks, and hostnames. We evaluate three networks of varying complexity, testing each prompt twenty-five times to determine the average response accuracy. Additionally, we assess prompts with different degrees of ambiguity for a complex network scenario, where the

TABLE III: Proposed scoring scheme for deliverable partitioning.

Type of Error	Penalty (%)
All IP's are not provided	5
All subnet masks are not provided	5
All hostnames not defined	5
All interfaces not defined	5
"Configure terminal" is missing	5
Routing commands not provided	5
Router "exit" keyword is missing	5
Router "end" keyword is missing	5
Incorrect IP address	5
Incorrect subnet mask	5
Incorrect hostname	5
Incorrect routing command	5
Incorrect routing protocol used	5
Incorrect number of nodes defined	5
Invalid topology constructed	5
Incorrect syntax	5
Unnecessary commands	5
Spelling error	5
Asked for user input	5
Combined all commands in one prompt	5

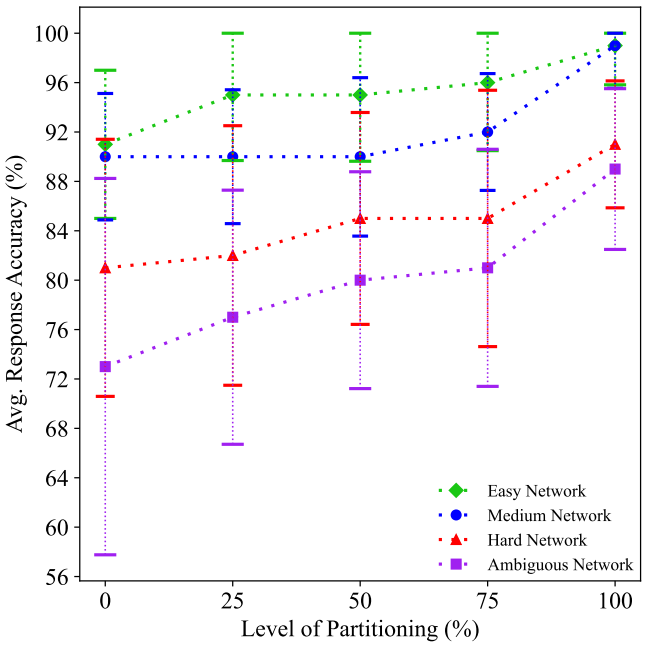


Fig. 3: Average response accuracy of network partitioned prompts with Copilot.

LLM is required to infer configuration parameters and make controlled design decisions. Microsoft's Copilot is used for the breadth of our evaluation. Figure 2 illustrates the topology of the evaluated medium-complexity network. We begin by providing a fully descriptive prompt for the target network and then apply the partitioning methodology described in Table I to divide the task into one or more prompts. Each response is subsequently evaluated using our proposed scoring policy.

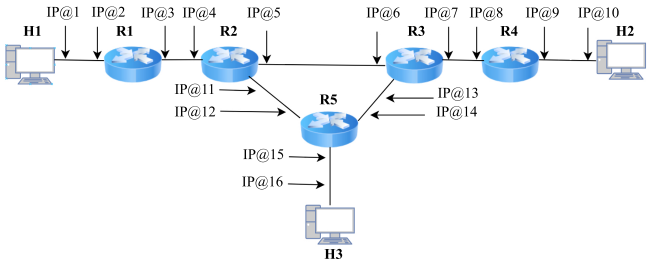


Fig. 2: Example of an evaluated network topology.

Figure 3 shows the results obtained with network partitioning, highlighting the performance improvement achieved by partitioning at the network-element level. The upward trend in average response accuracy is further supported by a noticeable drop in standard deviation, indicating more stable and consistent outputs—less affected by the intrinsic randomness of typical LLMs. Notably, this stability persists even with the introduction of ambiguity, demonstrating the scalability and robustness of the partitioning approach. These promising results are especially significant given that they are achieved using a general-purpose LLM like Microsoft Copilot, which, despite being trained on extensive datasets, may lack specific focus on networking or communications. This suggests that finer-grained partitioning can enhance performance in domain-specific tasks by minimizing the model's dependence on broad, unrelated knowledge.

B. Experiment 2: Comparison of Leading LLMs

Our partitioning methodology can be seamlessly adapted to a wide range of LLMs, enabling analysis of its cross-model performance. From the many LLMs available, we conduct extensive testing on three widely recognized models: Microsoft's Copilot, High-Flyer's DeepSeek, and OpenAI's ChatGPT. To ensure a fair comparison, we construct a consistent set of partitioned prompts for a single network and evaluate their performance across the three LLMs, running twenty-five iterations per model. As in previous experiments, we apply the response scoring policy described in Table II to assess correctness and consistency.

Figure 4 shows that Microsoft's Copilot outperforms the other models, with DeepSeek and ChatGPT following, highlighting Copilot's inherent strength in analyzing shorter, more concise input prompts. Despite noticeable performance differences among the models, each demonstrates a satisfactory average response accuracy. This suggests that the partitioning methodology is broadly adaptable and capable of consistently generating strong responses across different LLMs.

C. Experiment 3: The Impact of Deliverable Partitioning

To thoroughly explore different forms of partitioning, we next examine deliverable partitioning, which is a strategy aimed at reducing an LLM's reliance on multi-part output generation. In this experiment, we compare the response accuracy of partitioned versus non-partitioned prompts using ChatGPT-4o across three levels of ambiguity. We define ambiguity as the absence of necessary information, requiring the LLM to

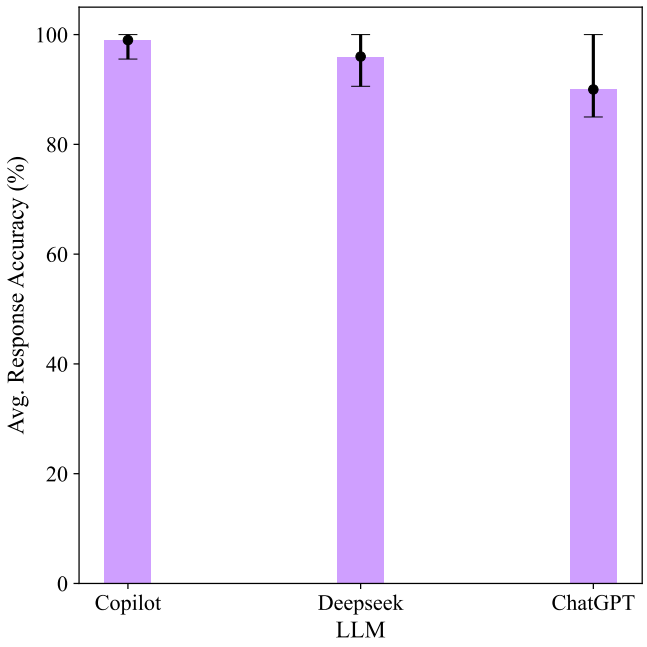


Fig. 4: Average response accuracy of partitioned prompts with various LLMs.

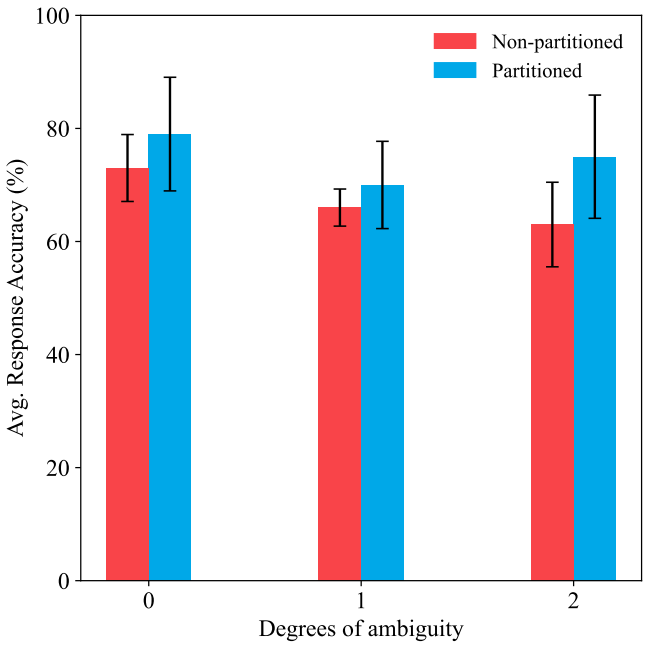


Fig. 5: Average response accuracy of deliverable-partitioned prompts with varying degrees of ambiguity.

infer the prompt's intent. At ambiguity level 0, all relevant information is provided; at level 1, IP address definitions are omitted; and at level 2, interface names are also omitted. We request two deliverables: (1) a set of router configuration commands to initialize all network elements, and (2) routing commands to establish end-to-end connectivity. Using our scoring methodology, outlined in Table III, we also perform 25 iterations per prompt to gather statistical insights and identify accuracy trends.

Figure 5 shows the results of the experiments, highlighting the ability of the prompt-partitioning methodology to improve the overall response accuracy. When reducing the deliverables requested, the LLM has a better chance of solving the problem statement and better recognizes the intent of the end-user, effectively providing a greater average response accuracy.

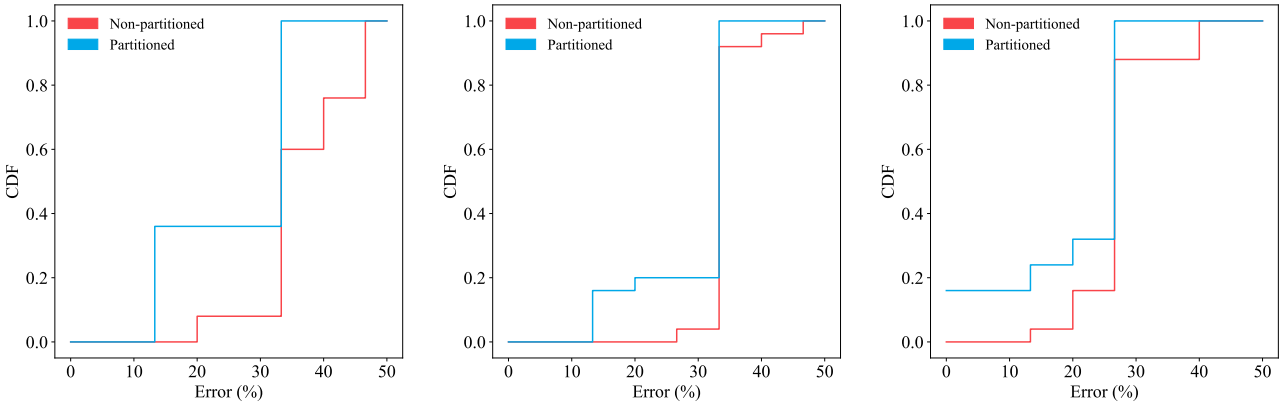
The figure highlights that prompts with no ambiguity—i.e., with all parameters fully specified (100% specificity), achieve an average accuracy improvement of approximately 5% when partitioned. However, this comes with higher variability, as indicated by a larger standard deviation. In contrast, prompts with one degree of ambiguity show a slightly lower average accuracy but exhibit reduced variability, suggesting more consistent, though slightly less accurate, responses.

Interestingly, in the case of two degrees of ambiguity, where the LLM must infer more information, the partitioned approach yields higher average accuracy than the previous two scenarios. Meanwhile, the non-partitioned approach performs worse than before. This improvement in the partitioned case likely stems from the model's initial parameter selections guiding subsequent ones more effectively. However, this gain in accuracy is accompanied by an increase in standard deviation, reflecting greater variability in the results.

These results indicate a complex and random process to respond to users' intents by an LLM. They indicate that leaving the LLM to make more selections may provide some accurate results but at the same time more variability of responses. However, if the ambiguity in prompts is measured, there is more consistency in the responses, which may indicate more predictable results. It is important to note that the value of LLMs in network configuration and design is anchored in using a high degree of ambiguity as that permits a higher degree of automation.

We also identify the quality of LLM responses by analyzing not only the average accuracy but the frequency in which high-quality responses are obtained by analyzing the cumulative distribution function (CDF) of the accuracy of the prompts. These CDF are presented as a function of the degree of ambiguity, which is defined as the number of parameters not specified with the prerogative of testing an LLM to take *educated* decisions on network design, as showcased in Fig. 6. While it boasts better results regardless of the level of ambiguity involved, partitioning excels at higher ambiguity degrees, which stems from the greater variation in the response pool of the LLM at higher degrees of ambiguity. Reducing the number of deliverables reduces the size of the potential response pool, further improving the overall solution capabilities of the model.

Fig. 6.a shows the CDF of prompts with 100% specificity (no ambiguity) for the partitioned and non-partitioned approaches. As the figure shows, the partitioned approach accumulates higher responses faster than the non-partitioned approach. In fact, the graph shows that the highest accuracy reached by the non-partitioned approach is 80% while that of the partitioned approach is about 87%.



(a) Prompt with no ambiguity **(b)** Prompt with 1 degree of ambiguity **(c)** Prompt with 2 degrees of ambiguity

Fig. 6: CDF of the response accuracy.

For cases with 1 degree of ambiguity (one element of design left to the LLM to select, such as IP addresses), as shown in Figure 6.b, the CDF shows again that the partitioned approach achieves cases with higher accuracy than the non-partitioned one, and also that in general, most cases of the partitioned approach achieve higher accuracy than the non-partitioned approach.

For cases with 2 degrees of ambiguity, Figure 6.c, namely the IP addresses and the routing information, the results show that the partitioned approach again produces responses with higher accuracy than those from the non-partitioned one. In fact, this figure shows that the partitioned approach achieves 75% accuracy and higher, while the non-partitioned approach achieves 60% accuracy and up to 85%.

V. Conclusions

We proposed partitioning of the prompting process as a method to increase the correctness of LLM responses to design and configure data networks. The partitioning of the process focuses on simplifying the design intent (deliverable) to provide specific and correct answers. The objective is to streamline the LLM responses to the network equipment for rapid configuration. We proposed multiple scoring policies to evaluate the configuration commands and to identify errors for improvements. The prompting deliverables partition method was tested with non-partitioned network descriptions and compared with prompts that partition the network description. The results show that deliverable partitioning achieves higher average accuracy than the non-partitioning approach and also offers more consistent results. These results also provide an insight into using LLMs for making design choices and inferring network parameters rather than using them for well-defined tasks.

References

[1] Z. Ben Houidi and D. Rossi, "Neural language models for network configuration: Opportunities and reality check," *Computer Communications*, vol. 193, p. 118–125, Sep. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2022.06.035

[2] A. Çelen, G. Han, K. Schindler, L. V. Gool, I. Armeni, A. Obukhov, and X. Wang, "I-Design: Personalized LLM Interior Designer," 2024. [Online]. Available: https://arxiv.org/abs/2404.02838

[3] V. Komanduri, S. Estropia, S. Allessio, G. Yerdelen, T. Ferreira, G. Palomino-Roldan, Z. Dong, and R. Rojas-Cessa, "Fine tunning LLM prompts for automating network design and configuration," in *2025 IEEE International Conference on Artificial Intelligence in Information and Communications*, 2025, pp. 1–6.

[4] D. Donadel, F. Marchiori, L. Pajola, and M. Conti, "Can llms understand computer networks? towards a virtual system administrator," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*. IEEE, 2024, pp. 1–10.

[5] V. Komanduri, C. Wang, and R. Rojas-Cessa, "Comparing link sharing and flow completion time in traditional and learning-based TCP," in *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*, 2024, pp. 167–172.

[6] R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do LLMs need to Synthesize Correct Router Configurations?" 2023. [Online]. Available: https://arxiv.org/abs/2307.04945

[7] M. Besta, L. Paleari, A. Kubicek, P. Nyczyk, R. Gerstenberger, P. Iff, T. Lehmann, H. Niewiadomski, and T. Hoefler, "CheckEmbed: Effective Verification of LLM Solutions to Open-Ended Tasks," 2024. [Online]. Available: https://arxiv.org/abs/2406.02524

[8] G. Sun, Y. Wang, D. Niyato, J. Wang, X. Wang, H. V. Poor, and K. B. Letaief, "Large Language Model (LLM)-enabled Graphs in Dynamic Networking," 2024. [Online]. Available: https://arxiv.org/abs/2407.20840

[9] 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.

[10] M.-V. Dumitru, V.-A. Bădoiu, A. M. Gherghescu, and C. Raiciu, "Generating P4 Dataplanes Using LLMs," in *2024 IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*, 2024, pp. 31–36.

[11] O. G. Lira, O. M. Caicedo, and N. L. S. da Fonseca, "Large Language Models for Zero Touch Network Configuration Management," 2024. [Online]. Available: https://arxiv.org/abs/2408.13298

[12] 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*, ser. ACM SIGCOMM '24. ACM, Aug. 2024, p. 661–678. [Online]. Available: http://dx.doi.org/10.1145/3651890.3672268

[13] B. Ifland, E. Duani, R. Krief, M. Ohana, A. Zilberman, A. Murillo, O. Manor, O. Lavi, H. Kenji, A. Shabtai, Y. Elovici, and R. Puzis, "GeNet: A Multimodal LLM-Based Co-Pilot for Network Topology and Configuration," 2024. [Online]. Available: https://arxiv.org/abs/2407.08249