Prompting Wireless Networks: Reinforced In-Context Learning for Power Control

Hao Zhou [*1] **Chengming Hu** [*1] **Dun Yuan** [1] **Ye Yuan** [1] **Di Wu** [2] **Xue Liu** [1] **Jianzhong (Charlie) Zhang** [3]

Abstract

To manage and optimize constantly evolving wireless networks, existing machine learning (ML)-based studies operate as black-box models, leading to increased computational costs during training and a lack of transparency in decision-making, which limits their practical applicability in wireless networks. Motivated by recent advancements in large language model (LLM)-enabled wireless networks, this paper proposes **ProWin**, a novel framework that leverages reinforced in-context learning to design task-specific demonstration **Pro**mpts for **Wi**reless **N**etwork optimization, relying on the inference capabilities of LLMs without the need for dedicated model training or fine-tuning. The task-specific prompts are designed to incorporate natural language descriptions of the task description and formulation, enhancing interpretability and eliminating the need for specialized expertise in network optimization. We further propose a reinforced in-context learning scheme that incorporates a set of advisable examples into task-specific prompts, wherein informative examples capturing historical environment states and decisions are adaptively selected to guide current decision-making. Evaluations on a case study of base station power control showcases that the proposed ProWin outperforms reinforcement learning (RL)-based methods, highlighting the potential for next-generation future wireless network optimization.

1. Introduction

The envisioned 6G network is expected to become increasingly complex, encompassing diverse application scenarios and advanced signal processing techniques such as mmWave and THz networks, reconfigurable intelligent surfaces, near-field communications, and movable antennas, among others (Zhang et al.). To optimize such constantly evolving networks, convex optimization-based methods commonly formulate a dedicated problem for each specific task and transform the objective functions or constraints into a convex form (Liu et al., 2024). On the other hand, although machine learning (ML)-based methods, such as reinforcement learning (Zhou et al., 2023; Burkart & Huber, 2021), require less stringent problem formulations and exhibit impressive performance across diverse tasks, the tedious model training and fine-tuning demand extensive computational resources, making them impractical for wireless networks, particularly in resource-constrained environments. Additionally, existing ML-based methods operate as black boxes, lacking interpretability and transparency in the decision-making process, which poses challenges in wireless network-related applications. Hence, the above challenges, including redundant problem formulation, computationally intensive demands, and lack of transparency, are calling for more advanced methods to optimize and manage next-generation wireless networks.

Generative AI (GenAI), particularly large language models (LLMs), has recently attracted considerable attention in the field of natural language processing, offering promising opportunities to develop LLM-enabled wireless networks across diverse tasks, including edge intelligence (Lin et al., 2023), semantic communication (Park et al., 2024; Chen et al., 2024), and network design (Qiu et al., 2024), among others. Specifically, in-context learning leverages the inference capabilities of LLMs by incorporating natural language-based demonstration prompts to guide task execution (Dong et al., 2022). By following task-specific descriptions within demonstration prompts, LLMs can identify the task type using knowledge embedded in pre-trained corpora and adopt effective task-solving strategies (Wies et al., 2023; Xue & Salim, 2023; Ouyang et al., 2022). Given the high computational efficiency without dedicated model training or fine-tuning, in-context learning presents a promising method in wireless networks, particularly in resource-constrained environments, by reducing the energy consumption and alleviating computational burden on net-

[*]Equal contribution [1]School of Computer Science, McGill University, Montreal, QC H3A 0E9, Canada. [2]School of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada. [3]Samsung Research America, Plano, Texas, TX 75023, USA.. Correspondence to: Hao Zhou <haozhou029@gmail.com>.

work resources. Moreover, in-context learning enables natural language-based task design and implementation through demonstration prompts, allowing operators to easily formulate problems using human language and instructions (Min et al., 2022). This user-friendly method holds great potential to reduce human effort and lower the need for specialized expertise in wireless networks. In-context learning also offers a distinct advantage in explainable decision-making process by generating natural language-based justifications for outputs (Bariah et al., 2023), thereby enhancing the transparency and interpretability of the reasoning process and enabling operators to clearly understand and manage wireless networks in the 6G era (Maatouk et al., 2024).

To this end, this paper proposes **ProWin**, a framework that leverages reinforced in-context learning to design task-specific demonstration **Pro**mpts for **Wi**reless **N**etwork optimization, eliminating the need for model training or fine-tuning. Compared to existing LLM-enabled wireless network studies (Su et al., 2024; Yan et al., 2025; Park et al., 2024; Qiu et al., 2024), our proposed ProWin addresses a more complex and dynamic network optimization problem, enabling the LLM to make informed decisions by learning from and adapting to varying network observations. The task-specific prompts are first designed to incorporate natural language descriptions of task description related to task objectives, definitions, and rules, making it efficient to formulate tasks while reducing the need for specialized knowledge in network optimization. To better optimize dynamically evolving network environments, we further propose a reinforced in-context learning scheme that adaptively embeds a set of advisable examples into task-specific prompts. Specifically, each example comprises a historical environment state, the decision made in that state, and the corresponding reward evaluated after the decision. Moreover, to effectively select relevant examples for the current task, we introduce state-based and ranking-based schemes for discrete and continuous states, respectively, ensuring that the most informative past examples are selected to provide valuable guidance for current decision-making. In this way, the LLM is expected to make decisions based on task-specific prompts embedded with relevant examples. The resulting decision, current environment state, and corresponding evaluated reward are then stored as a new example in an experience pool, serving as a reference for future decision-making. In this paper, we consider base station (BS) power control as a case study, which is a fundamental and critical optimization task in wireless networks that has been extensively studied with diverse algorithms, including convex optimization, game theory, and reinforcement learning, among others. To summarize, the **main contributions** of our work are provided as follows:

- We propose ProWin, a novel LLM-enabled method that designs task-specific demonstration prompts for wireless network optimization, eliminating the need for model training or fine-tuning, and enabling efficient task formulation with reduced reliance on specialized expertise in network optimization.
- We introduce a reinforced in-context learning scheme that adaptively selects and embeds a set of advisable examples into task-specific prompts, ensuring effective network optimization in dynamically evolving environments through guidance from the selected examples.
- We conduct comprehensive experiments on the case study of base station power control, demonstrating the consistent superiority of the proposed across ProWin various LLMs.

2. System Model

2.1. Problem Formulation

This section introduces a BS power control problem, serving as a case study to demonstrate the application of the proposed ProWin for wireless network optimization. Considering a BS with U_b users, the achievable data rate $C_{b,u}$ between BS b and user u is defined by (Zhou et al., 2022)

$$C_{b,u} = \sum_{k=1}^{K_b} d_k log(1 + \frac{p_{b,k} h_{b,k,u} \gamma_{b,k,u}}{\sum\limits_{b' \in B_{-b}} p_{b',k'} h_{b',k',u'} \gamma_{b',k',u'} + d_k N_0}), \quad (1)$$

where K_b is the total number of resource blocks (RBs) in BS b, d_k is the bandwidth of RB k, $p_{b,k}$ indicates the transmission power of BS b on RB k, $h_{b,k,u}$ defines the channel gain between BS b and user u on RB k, and N_0 is the noise power density. For the RB allocation, $\gamma_{b,k,u} \in \{0,1\}$ indicates whether RB k is allocated to the transmission for user u. For the interference, B_{-b} represent the set of adjacent BSs except for BS b, $p_{b',k'} h_{b',k',u'} \gamma_{b',k',u'}$ defines the inter-cell interference, and we assume orthogonal frequency-division multiplexing is applied to eliminate intra-cell interference.

This work aims to minimize the BS transmission power and meanwhile satisfy the average data rate constraint (Chiang et al., 2008):

$$\min_{P_b} \quad \sum\nolimits_{b \in B} P_b \quad (2)$$

$$\text{s.t.} \quad 0 \le P_b \le P_{max}, \quad (2a)$$

$$P_b = \sum\nolimits_{k=1}^{K_b} p_{b,k}, \quad (2b)$$

$$\sum\nolimits_{u=1}^{U_b} C_{b,u}/U_b \ge C_{min}, \quad (2c)$$

where P_b is the total transmission power of BS b and $P_b = \sum_{k=1}^{K_b} p_{b,k}$, $p_{b,k}$ has been defined in equation (1) as the transmission power of RB k, P_{max} is the maximum power, U_b is the total number of users, and C_{min} is the average achievable data rate constraint. We assume P_b is equally allocated to all RBs, and a proportional fairness method is used for RB allocation, which has been widely used as a classic approach. Then we can better focus on LLM features.

2.2. Language-based Power Control Task Description

Problem (2) has been extensively investigated in existing studies, but this work differs from previous works by presenting a unique view from the perspective of LLM-enabled network optimization. Instead of defining specific equations as in (2), here we use natural language to describe the optimization task: 1) Firstly, it inherently avoids the complexity of defining dedicated problem formulations, which is usually time-consuming; 2) Secondly, language-based task description is a user-friendly approach, and network operators can easily formulate the task without requiring any professional optimization-related knowledge.

In this power control case study, the task description involves "$Task\_goal$", "$Task\_definition$", and extra "$Rules$". The defined task description is shown below, which will further be used to prompt LLMs:

Task description for BS transmission power control

Task goal: You have a decision-making task for base station power control, and you need to select between 4 power levels from 1 to 4.
Task definition: You have to consider the specific user number of each case, which is the "base station user number".
Following are some examples $\{Example\_set\}$.
Now I will give you a new condition to solve, the current BS user number is $\{Num\_BS\_user\}$.
Rules: Now please select from "level 1", "level 2", "level 3", and "level 4" based on the above examples.

In particular, the $Task\_goal$ first specifies a "*decision-making task for base station power control*", and the goal is to "*select between 4 power levels*"[1]. Then the $Task\_definition$ introduces the environment states we need to consider. For example, this work assumes the total user numbers may change dynamically, and then the LLM has to consider the "*user number*" of each case. After that, the example set $\mathcal{E}_t$ is included by "*Following are some examples....*", and we provide a new condition for the LLM to solve with the current user number U_b. Finally, we set extra reply rules such as "*select from ... based on the above examples*", indicating the LLM to focus on decision-making.

[1]Here we select 4 power levels as an example, which can be changed to any number of levels

3. In-context Learning-based Optimization Algorithm

This section will introduce the proposed in-context learning algorithm, aiming to optimize power control by using natural language-based task descriptions in Section 2.2.

3.1. In-context Learning

In-context learning refers to the process that LLMs can learn from formatted natural language such as task descriptions and task solution demonstrations, to improve the performance on target tasks. In-context learning can be defined as (Dong et al., 2022)

$$D_{task} \times \mathcal{E}_t \times s_t \times \mathcal{LLM} \Rightarrow a_t, \tag{3}$$

where D_{task} is the task description and query, $\mathcal{E}_t$ is the set of examples at time t, s_t is the environment state at time t that is associated with the target task, $\mathcal{LLM}$ indicates the LLM model, and a_t is the LLM output. Here we expect the LLM can utilize the initial task description D_{task}, learn from the example set $\mathcal{E}_t$, and then make decision a_t based on current environment state s_t of the target task.

The LLM's in-context learning capabilities can be considered as implicit fine-tuning according to (Dai et al., 2022). LLMs will produce meta-gradients based on given examples $\mathcal{E}$ by forward computation, and then the meta-gradients are applied by using the attention mechanism to build an in-context learning model. Specifically, consider x as the input representation of a query token t, and $\mathbf{q} = W_Q x$ as the query vector in the attention mechanism. Then the attention of a head is formulated as

$$\begin{aligned} f_{ICL}(\mathbf{q}) =& \text{attention}(\mathbf{q}, K, V) \\ =& \text{softmax}(\frac{\mathbf{q}(W_K[\mathcal{Q};\mathcal{E}])^T}{\sqrt{d}})W_V[\mathcal{Q};\mathcal{E}] \end{aligned} \tag{4}$$

where W_Q, W_K and W_V represent the weight matrices of attention queries, keys, and values, respectively; d is the scaling factor; $\mathcal{Q}$ and $\mathcal{E}$ denote the input representations of query tokens and demonstration tokens, respectively, and $[\mathcal{Q};\mathcal{E}]$ is the concatenated matrix of $\mathcal{Q}$ and $\mathcal{E}$. For ease of qualitative analysis, we relax the standard attention to linear attention by removing the softmax operation and the scaling factor from equation (4):

$$\begin{aligned} f_{ICL}(\mathbf{q}) \approx& \mathbf{q}(W_K[\mathcal{Q};\mathcal{E}])^T W_V[\mathcal{Q};\mathcal{E}] \\ =& \mathbf{q}(W_K\mathcal{Q})^T W_V\mathcal{Q} + \mathbf{q}(W_K\mathcal{E})^T W_V\mathcal{E} \\ =& \tilde{f}_{ICL}(\mathbf{q}) \end{aligned} \tag{5}$$

We define $W_{ZSL} = (W_K\mathcal{Q})^T W_V\mathcal{Q}$ for ease of notations, since the W_{ZSL} parameters include the query token $\mathcal{Q}$ only.

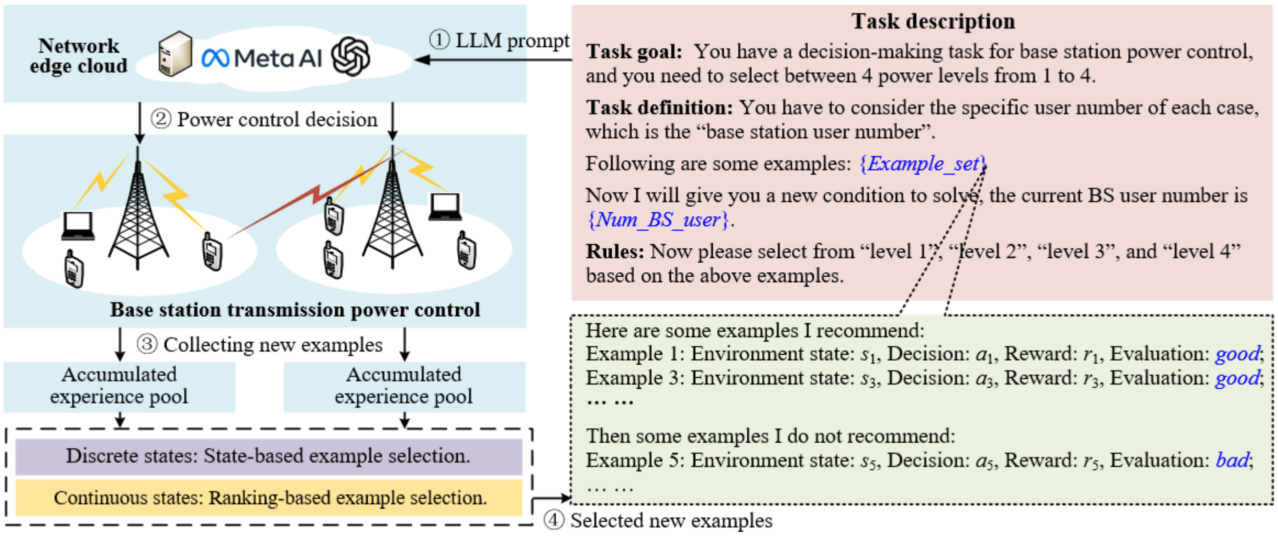

Figure 1. Overall design of the proposed LLM-enabled in-context learning for transmission power control.

Therefore, W_{ZSL} indicates the zero-shot learning case without examples. Then we can rewrite equation (5) as

$$\begin{aligned}\tilde{f}_{\text{ICL}}(\mathbf{q}) &= \mathbf{q}W_{\text{ZSL}} + \mathbf{q}\left(W_K\mathcal{E}\right)^T W_V\mathcal{E} \\ &= \mathbf{q}W_{\text{ZSL}} + \text{LinearAttention}\left(\mathbf{q}, W_K\mathcal{E}, W_V\mathcal{E}\right) \\ &= \mathbf{q}W_{\text{ZSL}} + \mathbf{q}\sum_i \left(\left(W_K E_i\right)^T W_V E_i\right) \\ &= \mathbf{q}W_{\text{ZSL}} + \mathbf{q}\sum_i \left(\left(W_K E_i\right) \otimes \left(W_V E_i\right)\right) \\ &= \mathbf{q}W_{\text{ZSL}} + \mathbf{q}\Delta W_{\text{ICL}} \\ &= \mathbf{q}\left(W_{\text{ZSL}} + \Delta W_{\text{ICL}}\right).\end{aligned} \tag{6}$$

Here $\Delta W_{\text{ICL}} = \sum_i \left((W_K E_i') \otimes (W_V E_i')\right)$, which indicates the updated weight when examples $E \in \mathcal{E}$ are provided. $\mathbf{q}\left(W_{\text{ZSL}} + \Delta W_{\text{ICL}}\right)$ also aligns with the weight updating of back-propagation algorithm. It proves that the examples $\mathcal{E}$ will affect the in-context learning weight ΔW_{ICL}, and therefore LLMs can learn from examples and generate replies.

3.2. Examples and Optimization Framework Design

The analyses in Section 3.1 show that examples are of great importance in in-context learning, which will directly affect the ΔW_{ICL} values. However, many network optimization problems have continuous environment states, e.g., adjusting the BS transmission power based on user-BS distance. Such cases mean that there may be an infinite number of examples, and therefore identifying the most relevant and useful examples becomes challenging. Here we define an example by

$$E = \{s, a, r(s,a)\}, E \in \mathcal{E}, \tag{7}$$

where s and a are environment state and decision, respectively. Inspired by reinforcement learning, we further define a reward value to evaluate the decision a by

$$r = P_{target} - P_b - \beta, \tag{8}$$

where P_{target} is a target power consumption, and P_b has been defined in problem (2) as the total power consumption of BS b. β is a penalty term, which is only applied when constraint (2c) is not satisfied. Then, r provides a comprehensive metric to evaluate the selected decision a under environment state s.

Fig.1 shows the overall design of the proposed in-context learning algorithm for transmission power control. Specifically, the above task description D_{task}, current environment state s_t, and selected examples $\mathcal{E}_t$ are integrated as input prompt as defined in equation (3), and then the LLM will generate a power control decision a_t based on s_t and the experiences in $\mathcal{E}_t$. Then, the decision a_t is implemented, the achieved data rate $C_{b,u}$ is collected, and the reward r_t is calculated as equation (8). $E_t = \{s_t, a_t, r_t(s_t, a_t)\}$ becomes a new example in the accumulated experience pool $\mathcal{E}_{pool}$ in Fig.1. After that, based on the next environment state s_{t+1}, a new example set $\mathcal{E}_{t+1}$ is selected, and the selected examples are inserted into the task description with s_{t+1}, becoming a new prompt for the LLM model to generate a_{t+1}.

3.3. State-based Example Selection for Discrete State Problems

Selecting appropriate examples is critical for in-context learning(Chen et al., 2023). For problems with discrete

Algorithm 1 Proposed In-context Learning-based Algorithm for Network Optimization

Input: Network parameters: BS and user locations, BS transmission power constraint P_{max}, user data rate constraint C_{min}. Optimization parameters: penalty term β, ϵ for epsilon-greedy policy, weighting factor τ. Initialize the experience pool $\mathcal{E}_{pool}$.
Designing input prompt: Task goals, task definition, and rules as in Section 2.2.
repeat
  if $rand(0,1) < \epsilon$ **then**
    Selecting transmission power level randomly.
  else
    if Discrete state problem **then**
      Given current state s, selecting relevant examples $\mathcal{E}_{relevant}$ using equation (9) from experience pool $\mathcal{E}_{pool}$.
    else if Continuous state problem **then**
      Calculating the $\mathcal{L}(E, s_{target})$ metric of all examples in $\mathcal{E}_{pool}$ using equation (11), and then selecting the top-K relevant examples $\mathcal{E}_{relevant}$.
    end if
    Inserting the selected examples $\mathcal{E}_{relevant}$ into the prompting template. Feeding the prompts to LLMs $D_{task} \times \mathcal{E}_t \times s_t \times \mathcal{LLM} \Rightarrow a_t$, and generating the selected power level a_t.
  end if
  Inserting the new example $E = \{s, a, r(s,a)\}$ into the accumulated experience pool $\mathcal{E}_{pool}$.
until Reaching the max number of iterations or the result converges.

environment states, relevant demonstrations can be easily identified by finding existing examples with the same states in the accumulated experience pool $\mathcal{E}_{pool}$. Considering a target task with environment state value s_{target}, the set of relevant examples can be identified by

$$\mathcal{E}_{relevant} = \left\{ E\{s, a, r(s,a)\} \middle| s = s_{target}, E \in \mathcal{E}_{pool} \right\} \tag{9}$$

where $\mathcal{E}_{pool}$ is the accumulated experience pool in Fig. 1. Given $\mathcal{E}_{relevant}$, we can easily select recommended examples with high reward, i.e., top-K examples, and inadvisable examples, e.g., examples with lower reward or violating the minimum data rate constraint.

In addition, we include a well-known epsilon-greedy policy to balance exploration and exploitation.

$$a = \begin{cases} \text{Random action selection}, & \text{if } rand < \epsilon; \\ \text{LLM-based decision-making}, & \text{else}, \end{cases} \tag{10}$$

where ϵ is a predefined value, and $rand$ is a random number between 0 and 1. Therefore, the random exploration in equation (10) can constantly explore new examples, and then the LLM model can learn from better relevant examples $\mathcal{E}_{relevant}$ to improve the performance.

3.4. Ranking-based Example Selection for Continuous State Problems

Compared with discrete-state problems, environments with continuous states can be much more complicated. For instance, when using average user-BS distance as an environment state for BS transmission power control with a target task s_{target}, it is unlikely to find a specific existing example $E\{s, a, r(s,a)\}$ with $s = s_{target}$, since s_{target} is a random number within the BS maximum coverage distance. This problem may be solved by discretizing the continuous states into some discrete values, but this may still lead to a large number of states or extra errors. To this end, we define a new metric $\mathcal{L}$ for example selection with continuous states:

$$\mathcal{L}(E, s_{target}) = r(s,a) - \tau ||s - s_{target}||, \tag{11}$$

where $\mathcal{L}(E, s_{target})$ is a comprehensive metric to evaluate the usefulness of $E = \{s, a, r(s,a)\}$ to the decision-making of s_{target}, and $||s - s_{target}||$ is the L^2 norm to define the distance between s and s_{target}. Equation (11) aims to jointly consider the reward and states of example E, and τ is a weighting factor to balance the importance of higher reward $r(s,a)$ and more similar states between s and s_{target}. Specifically, a higher reward $r(s,a)$ indicates that E includes a good action selection a under environment state s, and meanwhile lower $||s - s_{target}||$ value means the environment state s in E is more similar to s_{target}. Therefore, we use $\mathcal{L}(E, s_{target})$ as a comprehensive metric, and then the recommended and inadvisable examples can be selected similarly as in Section 3.3 by selecting the top-K examples with highest $\mathcal{L}(E, s_{target})$ value.

Finally, the proposed algorithm can be summarized as Algorithm 1. With the epsilon-greedy policy, the algorithm can try different actions by random exploration, finding better network optimization decisions. On the other hand, LLM can utilize the accumulated experience and examples to make good decisions, balancing exploration and exploitation. Such a design indicates that LLMs can constantly ex-

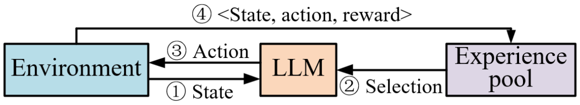

Figure 2. The overall procedure of the example-related scheme.

plore the network environment and improve their decisions iteratively. Compared with other ML-enabled optimization techniques, the proposed algorithm has no need to update LLM model parameters. Therefore, it is more efficient with much lower complexity.

3.5. Computational Complexity Analyses

Fig. 2 summarizes the overall procedure of example-related schemes. In particular, the LLM will receive the state from the environment, and then use the examples provided by the experience pool to select actions such as the transmission power level. The implementation results will become a new example for the pool. Meanwhile, no additional computational cost is incurred for example selection, as each new example is simply appended to the accumulated experience pool after implementation. Secondly, for example selection in discrete state problems, it is easy to search the experience pool to identify $s = s_{target}$. For continuous states, we calculate the $\mathcal{L}(E, s_{target})$ metric for all examples in the pool, and then select the best examples accordingly. Therefore, the cost of example selection follows a linear complexity. Finally, note that the LLM inference time is affected by model architecture, hardware constraints, and task types, and it can also be further optimized by quantization, sparsity exploitation, and architectural innovations.

4. Performance Evaluation

4.1. Simulation Settings

We consider three adjacent small base stations (SBSs); the user number of each SBS randomly changes from 5 to 15, and the SBS's coverage is 20 meters. The channel gain applies 3GPP urban network models, and 2 cases are evaluated: **Case I**: Discrete states defined by user numbers of each SBS; **Case II**: Continuous states defined by average user-SBS distance, which represents 2 kinds of network optimization problems.

Then, the simulation considers 2 main approaches:
1) LLM-based method includes 3 models: Llama3-8b-instruct, Llama3-70b-instruct, and GPT-3.5 turbo. Llama3-8b is a small-scale LLM, while Llama3-70b and GPT-3.5 turbo are large models. Using LLM models with various sizes can better evaluate the capabilities of our proposed algorithms(Oh et al., 2024). The first 40% episodes in the simulation are exploration phases, while the rest of the episodes

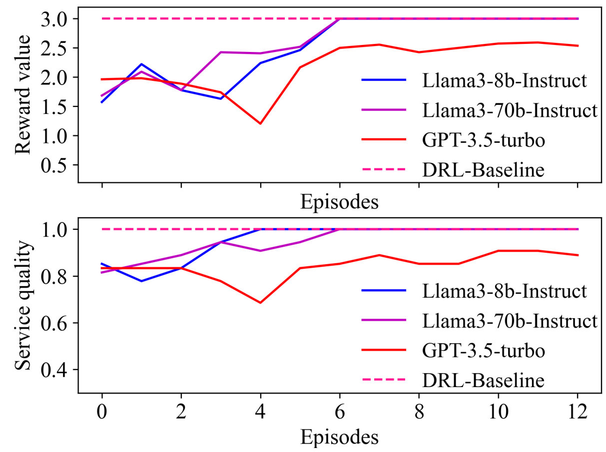

Figure 3. Discrete state space: System reward and service quality comparison of various LLMs.

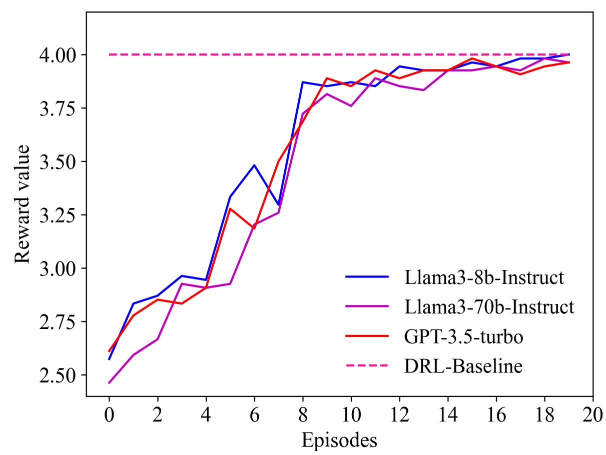

Figure 4. Continuous state space: System reward comparison of various LLMs.

are the exploitation phase.
2) DRL-based method: with dedicated model training, here we consider DRL as an optimal baseline since its capability has been demonstrated in many existing studies (Zhang & Liang, 2020; Zhou et al., 2022). The Markov decision process (MDP) for deep Q-learning is: states can be easily defined by considering discrete/continuous states as introduced above, actions indicate the BS transmission power levels, and rewards are defined as equation (8).

4.2. Simulation Results

Fig. 3 to 8 show the simulation results and comparisons.

1) Discrete State Problems: Firstly, for discrete state problems, Fig. 3 presents the system reward and service quality of different LLMs. One can observe that both Llama3 LLMs achieve a comparable reward and service quality as the DRL

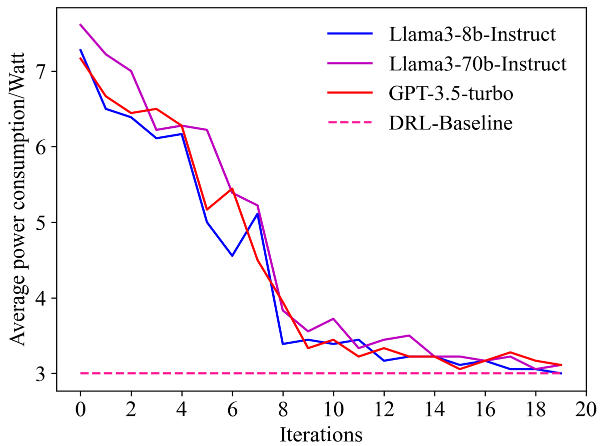

Figure 5. Continuous state space: Power consumption comparison of various LLMs.

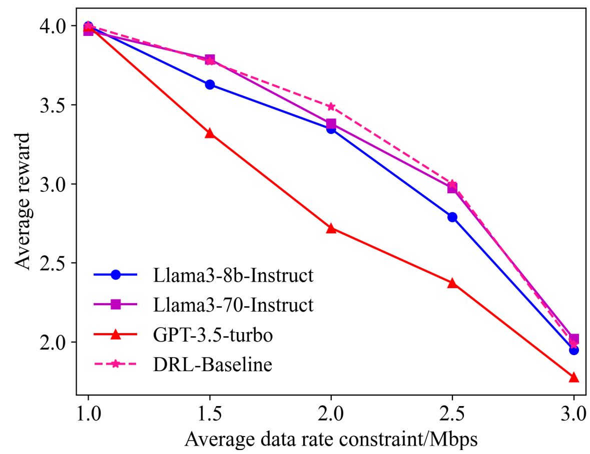

Figure 6. Continuous state space: Average reward comparison under different data rate constraints.

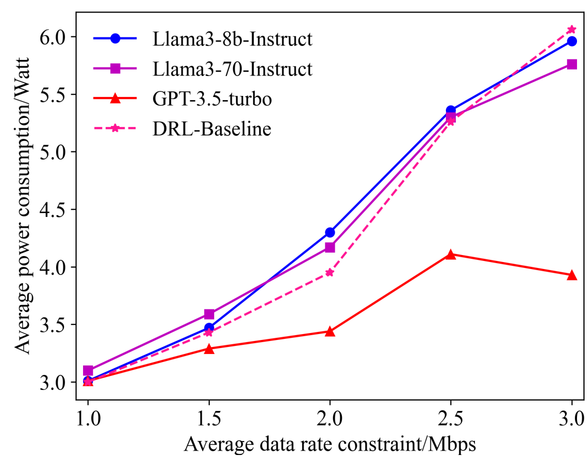

Figure 7. Continuous state space: Average power consumption comparison under different data rate constraints.

baseline, while GPT-3.5 shows a lower reward and service quality. Fig. 3 demonstrates that the proposed in-context learning algorithm can provide satisfactory performance for problems with a limited number of environment states.

2) Continuous State Problems: Then, we consider more complicated scenarios with continuous states defined by the average user-BS distance. Fig. 4 and 5 show that all LLM models achieve higher rewards and lower power consumption as the number of episodes increases and finally converge to stable values, e.g., reward value 4 and 3 Watts average power consumption. Specifically, with the epsilon-greedy policy, the LLM can randomly explore different actions and meanwhile make optimal decisions based on existing knowledge. The results demonstrate that LLMs can learn from previously accumulated examples and then improve their performance on target tasks. Such an iterative optimization approach is a crucial skill in addressing many real-world problems.

3) Changed Environment and Simulation Settings: In addition, we observe the algorithm's performance under different minimum data rate constraints. Fig. 6, 7, and 8 present the average reward, power consumption, and service quality, respectively. Here, every value in the following Fig. 6 to 8 is obtained by taking the average performance of converged episodes of corresponding LLMs as in Fig. 4 and 5. As expected, the simulation results show that increasing the minimum data rate constraint leads to lower reward, lower service quality, and higher power consumption. They demonstrate that the proposed in-context learning can adapt to different optimization settings and then adjust their policies to improve the performance of target tasks. Under various environment requirements, the proposed technique can maintain a reasonable performance compared with the existing baseline DQN.

4) Scalability of the Proposed Algorithm: Fig. 9 evaluates the scalability of the proposed algorithm by increasing network size, e.g., the number of optimized BSs. It shows that the proposed techniques can maintain satisfactory performance with increasing base station numbers. LLM-based techniques have the potential for large-scale network management and optimization. The huge number of parameters of LLMs allows for large-scale information extraction, input and processing, and then generates the desired output.

5) Effect of the number of examples: In Fig. 10, we evaluated the system performance with enlarged state space and changing number of examples in the prompt. Firstly, one can observe that increasing the number of examples can constantly improve the average reward. However, such improvement becomes less obvious when plenty of examples are provided. On the other hand, increasing the state space means that more examples are needed in the prompt

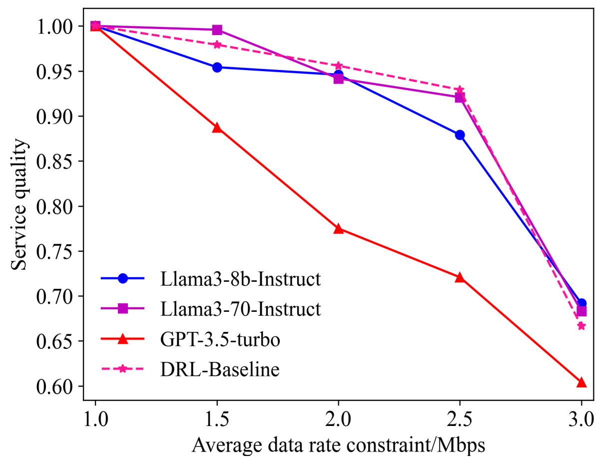

Figure 8. Continuous state space: Average service quality comparison under different data rate constraints.

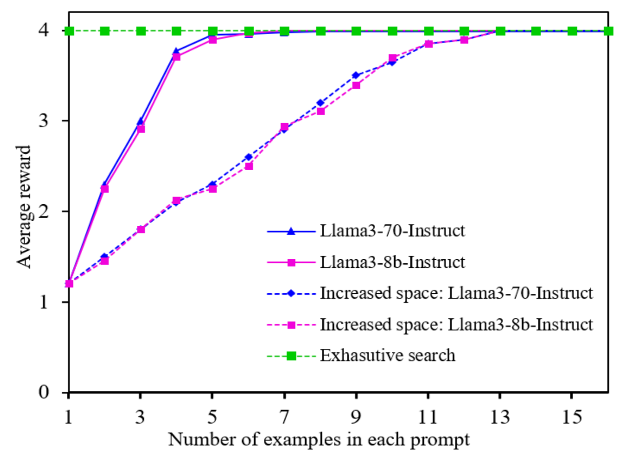

Figure 10. Reward performance with increasing number of examples and larger spaces.

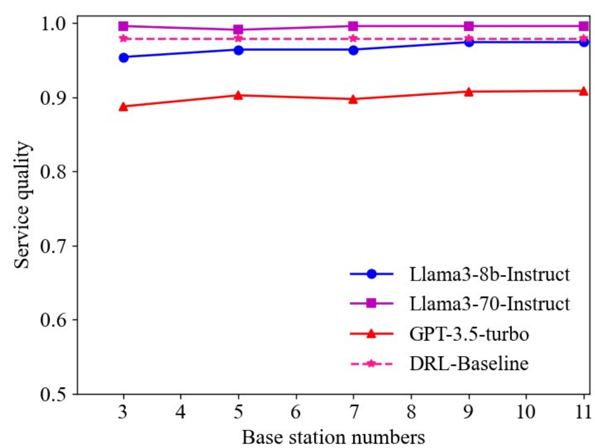

Figure 9. Continuous state space: Average service quality comparison with increasing number of BSs.

to achieve a satisfactory performance, e.g., more references and experience are needed to make proper decisions. However, it is worth noting that the overall performance is still constantly improving by increasing the number of provided examples, and it finally achieves a comparable performance as the exhaustive search method.

In summary, the above simulation results from Fig. 3 to Fig. 8 demonstrate that the proposed in-context learning technique can learn from previous explorations, optimize network performance iteratively, and adapt to different network environments. It achieves comparable performance as existing benchmarks DQN, and avoids the complexity of dedicated model parameter updating. In-context learning is considered a promising technique for future network optimization and management. The simulation results demonstrate that the algorithm's performance is closely related to the capabilities of specific LLMs. For instance, Llama3 represents state-of-the-art LLM designs, while GPT-3.5 is an early LLM model. Therefore, it is reasonable that Llama3-8b and Llama3-70b maintain comparable performance as the DRL baseline, while GPT-3.5 turbo presents a worse performance in different tasks. It highlights the importance of selecting appropriate LLMs to handle various tasks.

5. Conclusion

LLM is a promising technique for future wireless networks, and this work proposes an LLM-enabled in-context learning algorithm for BS transmission power control. The proposed algorithm can handle both discrete and continuous state problems, and the simulations show that it achieves comparable performance as conventional DRL algorithms. This work demonstrates the great potential of in-context learning for handling network management and optimization problems. In the future, we will consider more complicated network optimization tasks and involve more control variables and decisions.

References

Bariah, L., Zou, H., Zhao, Q., Mouhouche, B., Bader, F., and Debbah, M. Understanding telecom language through large language models. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pp. 6542–6547. IEEE, 2023.

Burkart, N. and Huber, M. F. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

Chen, J., Chen, L., Zhu, C., and Zhou, T. How many demonstrations do you need for in-context learning? *arXiv preprint arXiv:2303.08119*, 2023.

Chen, Z., Yang, H. H., Chong, K. F. E., and Quek, T. Q. Personalizing semantic communication: A foundation model approach. In *2024 IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 846–850. IEEE, 2024.

Chiang, M., Hande, P., Lan, T., Tan, C. W., et al. Power control in wireless cellular networks. *Foundations and Trends® in Networking*, 2(4):381–533, 2008.

Dai, D., Sun, Y., Dong, L., Hao, Y., Ma, S., Sui, Z., and Wei, F. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*, 2022.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., and et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Lin, Z., Qu, G., Chen, Q., Chen, X., Chen, Z., and Huang, K. Pushing large language models to the 6G edge: Vision, challenges, and opportunities. *arXiv preprint arXiv:2309.16739*, 2023.

Liu, Y.-F., Chang, T.-H., Hong, M., Wu, Z., Man-Cho So, A., Jorswieck, E. A., and Yu, W. A survey of recent advances in optimization methods for wireless communications. *IEEE Journal on Selected Areas in Communications*, 42(11):2992–3031, 2024. doi: 10.1109/JSAC.2024.3443759.

Maatouk, A., Piovesan, N., Ayed, F., De Domenico, A., and Debbah, M. Large language models for telecom: Forthcoming impact on the industry. *IEEE Communications Magazine*, 2024.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Oh, S., Kim, J., Park, J., Ko, S.-W., Quek, T. Q., and Kim, S.-L. Uncertainty-aware hybrid inference with on-device small and remote large language models. *arXiv preprint arXiv:2412.12687*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Park, J., Ko, S.-W., Choi, J., Kim, S.-L., Choi, J., and Bennis, M. Towards semantic mac protocols for 6g: From protocol learning to language-oriented approaches. *IEEE BITS the Information Theory Magazine*, 2024.

Qiu, K., Bakirtzis, S., Wassell, I., Song, H., Zhang, J., and Wang, K. Large language model-based wireless network design. *IEEE Wireless Communications Letters*, 2024.

Su, Y., Yan, N., and Deng, Y. Federated llms fine-tuned with adaptive importance-aware lora. *arXiv preprint arXiv:2411.06581*, 2024.

Wies, N., Levine, Y., and Shashua, A. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36:36637–36651, 2023.

Xue, H. and Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023. doi: 10.1109/TKDE.2023.3342137.

Yan, N., Su, Y., Deng, Y., and Schober, R. Federated fine-tuning of llms: Framework comparison and research directions. *arXiv preprint arXiv:2501.04436*, 2025.

Zhang, L. and Liang, Y.-C. Deep reinforcement learning for multi-agent power control in heterogeneous networks. *IEEE Transactions on Wireless Communications*, 20(4): 2551–2564, 2020.

Zhang, Z., Xiao, Y., Ma, Z., Xiao, M., Ding, Z., Lei, X., Karagiannidis, G. K., and Fan, P. 6g wireless networks: Vision, requirements, architecture, and key technologies. *IEEE vehicular technology magazine*, 14(3):28–41.

Zhou, H., Erol-Kantarci, M., and Poor, H. V. Learning from peers: Deep transfer reinforcement learning for joint radio and cache resource allocation in 5G RAN slicing. *IEEE Transactions on Cognitive Communications and Networking*, 8(4):1925–1941, 2022.

Zhou, H., Erol-Kantarci, M., Liu, Y., and Poor, H. V. A survey on model-based, heuristic, and machine learning optimization approaches in ris-aided wireless networks. *IEEE Communications Surveys & Tutorials*, 2023.

A. Input Prompt and LLM Replies Analyses

The following presents some supplementary experiment results and discussions, including detailed input prompts and different LLMs' replies, DRL algorithm design and implementation comparisons. Below is our designed prompt input to LLMs.

Prompt input of LLMs for BS power control

You have a decision-making task for Base Station power control.
Task goal: You need to select between 4 power levels from 1 to 4.

Task description: You have to consider the specific average user distance of each case.

Following are some good examples I recommend:

Example 49: Average user distance to the Base station: 8.78, your selected power level: 1.0, reward: 4.0, evaluation = good.

Example 85: Average user distance to the Base station: 9.24, your selected power level: 1.0, reward: 4.0, evaluation=good.

Example 58: Average user distance to the Base station: 9.19, your selected power level: 1.0, reward: 4.0, evaluation=good.

Example 67: Average user distance to the Base station: 9.15, your selected power level: 1.0, reward: 4.0, evaluation=good.

Example 13: Average user distance to the Base station: 9.15, your selected power level: 1, reward: 4, evaluation=good.

Example 49: Average user distance to the Base station: 8.96, your selected power level: 1.0, reward: 4.0, evaluation=good.

Following are some bad examples I do not recommend:

Example 21: Average user distance to the Base station: 10.05, your selected power level: 3, reward: 2, evaluation=bad.

Example 40: Average user distance to the Base station: 9.28, your selected power level: 4, reward: 1, evaluation=bad.

Example 120: Average user distance to the Base station: 9.9, your selected power level: 3, reward: 2, evaluation=bad.

Example 12: Average user distance to the Base station: 9.8, your selected power level: 3, reward: 2, evaluation=bad.

Example 39: Average user distance to the Base station: 10.02, your selected power level: 2, reward: 3, evaluation=bad.

Now I will give you a new condition to solve: Average user distance to the Base station: 8.72.

Please select from "level 1", "level 2", "level 3", and "level 4" based on the above examples. Please explain the reasons for your selection.

Reply Sample 1 of Llama3-8b instruct.

Based on the provided examples, I notice that when the average user distance is around 9.2, the recommended power level is consistently level 1. This is evident from examples 85, 58, 67, and 13, where the average user distance is between 9.15 and 9.24, and the selected power level is always 1.

Given the new condition: Average user distance to the Base station: 9.22, I will select 'level 1' as the recommended power level. My reasoning is that the average user distance is very close to the distances in the good examples (9.15-9.24), and the recommended power level in those cases was always level 1. Therefore, I believe that level 1 is the most suitable choice for this condition.

I do not recommend selecting higher power levels (2, 3, or 4) because they were not chosen in the good examples, and selecting them may result in a lower reward or evaluation, as seen in the bad examples.

Reply Sample 2 of Llama3-8b instruct.

Based on the provided examples, I notice that when the average user distance to the Base station is relatively short (around 8-9 units), the recommended power level is always 1.0, which corresponds to a "good" evaluation.
In the new condition, the average user distance to the Base station is 8.72, which is even shorter than the distances in the "good" examples. Therefore, I will select power level 1.0 for this condition.
I reason that since the average user distance is relatively short, it's likely that the users are closer to the Base station and require less power to maintain a good connection. Power level 1.0 has been consistently recommended in similar scenarios, so I believe it's the most suitable choice for this condition as well.

Reply Sample 3 of GPT-3.5 Turbo.

Based on the provided examples and conditions:

- Power level 1 (low power) was used in all the good examples where the average user distance ranged from 8.78 to 9.24, resulting in a reward of 4.0 in each case.

- Power levels 2, 3, and 4 (higher power levels) were used in the bad examples where the average user distance was greater than 9.28, leading to lower rewards ranging from 1 to 3.

Given that the new condition has an average user distance of 8.72, which falls within the range of the distances in the good examples, it would be advisable to select **level 1** for the power control. This choice is based on the pattern observed in the examples where lower power levels were more effective for shorter user distances, resulting in higher rewards and positive evaluations.

By selecting level 1 for this new condition, you are likely to achieve a good performance outcome similar to the ones observed in the good examples provided.

The above reply samples demonstrate that LLMs have the following capabilities:

- **LLMs can indeed learn from examples and then solve the target tasks**. For instance, all three reply samples start by "*Based on the provided examples, I notice that....*" and "*Based on the provided examples and conditions*". Sample 1 also provides detailed example numbers: "*This is evident from examples 85, 58, 67, and 13*". All three models immediately frame their decision around the provided examples ("Based on the provided examples…"), showing they can ingest few-shot demonstrations and abstract a decision rule. Even Llama3-8B (Sample 1) pulls out specific indices (85, 58, 67, 13), evidencing a capacity to reference and align with particular data points.

- **LLMs can clearly understand the relationship between control variables, environment states, and rewards in wireless network optimization problems.** For example, in reply sample 3 of GPT-3.5 turbo, it first analyzes the selection of power level 1 and 2 , 3, 4, then explains the *"Given that the new condition has an average user distance of..."*, and finally make the decision by *"it would be advisable to select **level 1** for the power control"*. Each reply articulates how the state variable (user distance) maps to the control variable (power level) and ultimately to the reward. GPT-3.5 Turbo (Sample 3) explicitly contrasts the "good" vs. "bad" examples based on distance thresholds before recommending level 1, underscoring an understanding of the underlying optimization structure.

- **LLM can provide useful explanations for their decisions.** One of the key advantages of LLMs is to provide detailed explanations of their selections. For instance, sample 2 shows that "*I reason that since the average user distance is relatively short, it's likely that the users are closer to the Base station and require less power to maintain a good connection*". Such a capability is crucial when facing more complicated scenarios, providing useful references for human understanding. This is also far beyond the capabilities of existing ML techniques. Beyond selecting an action, the models generate coherent justifications. Sample 2's justification ("users are closer…require less power") demonstrates how LLMs can produce intuitive, domain-relevant reasoning in human-readable form—an advantage over opaque "black-box" predictors.

- **Structured presentation and clarity.** Finally, GPT-3.5's bullet-point format lays out observations, contrasts, and conclusions in digestible steps, highlighting the models' ability to organize complex decision logic in a clear, hierarchical manner—which is essential for system-level reporting and auditability.

B. Potential Issues and Hallucination

We also observed other replies from the Llama3-8b instruct model, and noticed some potential issues. For instance, the text box below shows the LLM replies when given another task, in which the average user distance is 8.72.

Similar to previous replies, the LLM also clearly understands the key is to "*the relationship between the average user distance and the selected power level*". Then it analyzes the good and bad examples provided by "*From the good examples*" and "*From the bad examples*". After that, it compares the target condition (average user distance is 8.72) with the good examples by "*the average user distance to the Base station is 8.72, which is closer to...*". Finally, it explains the reasons why selecting power level 1, e.g., "*the average user distance is relatively close to the Base station, a lower power level (1) would be...*".

Although the LLM model made a good decision and obtained a satisfactory reward, it is worth noting that there is a problem when analyzing the distance. It claims that "*when the average user distance is farther away, e.g., 7.76, 8.12,*". However, 7.76 and 8.12 are obviously closer than the above distance of 8.72.

On the one hand, this experiment shows that hallucination is still one of the main problems for LLM implementations; on the other hand, in this task, we also noticed the importance of "good examples", which have a great impact on LLM's decision-making. For example, the 4th paragraph demonstrates that LLM considers good examples as critical references when selecting the power level.

Problematic Replies of Llama3-8b instruct.

Based on the provided examples, I will analyze the relationship between the average user distance and the selected power level.

From the good examples, I notice that when the average user distance is close to 8.78-8.94, the selected power level is always 1.0, which results in a high reward (4.0) and a good evaluation.

From the bad examples, I see that when the average user distance is farther away (e.g., 7.76, 8.12, or 9.71), the selected power level is higher (3 or 4), which results in a lower reward (1 or 2) and a bad evaluation.

Considering the new condition, the average user distance to the Base station is 8.72, which is closer to the range of the good examples (8.78-8.94). Therefore, I recommend selecting power level 1.

My reason is that since the average user distance is relatively close to the Base station, a lower power level (1) would be more suitable to maintain a good connection and achieve a high reward.