

Explaining Robot Failures in ROS using Parameter-Efficient Fine-Tuning

Ella Scheltinga and Chris Pek

Abstract—Autonomous robots are increasingly improving at performing a navigation task, however they will likely fail at some point or not perform as intended due to uncertainties or unforeseen situations in the real world. In such scenarios, explaining the robot’s behavior to humans is crucial to build trust and resolve potential issues. Recently, large language models (LLMs) have shown great potential in analyzing robot log data, e.g., obtained in ROS, and providing users with useful explanations. Yet, these models can often not consistently generate answers with high quality. In this study, we investigate the use of parameter-efficient fine-tuning (PEFT) to improve the explanations generated by LLMs. We use commercially available ROS navigation stacks for a TIAGo robot and present a fine-tuning framework for ROS log messages. Our preliminary results with state-of-the-art LLMs show promising improvement after fine-tuning to explain failures in navigation tasks. We are planning to conduct a user study to investigate the extent to which fine-tuning can lead to higher consistency and quality in explanations.

I. INTRODUCTION

As robots become increasingly integrated into our daily lives, and we encounter robot failures more frequently, the importance of explainability in human-robot interaction (HRI) becomes evident. Explainability fosters understanding between the user and a robot, contributing to building a trustworthy relationship between humans and robots. One effective approach is generating explanations that improve the user’s understanding about the behavior of the robot and why failures may have occurred.

Consider a simple navigation task, as shown in Figure 1 where a TIAGo robot is instructed to move along three waypoints while avoiding obstacles. TIAGo reaches waypoints 1 and 2, but not waypoint 3 and no path is shown towards this waypoint. Explainability systems aim to provide explanations for this. Ideally, such systems should provide answers to questions such as “Is there a feasible path to waypoint 3?”, and “Why is there no feasible path possible?”. In this situation, a path towards waypoint 3 is not feasible because it is located within a known obstacle.

While the literature lacks consensus on the precise definition of the concept of explainability [1], [2], there is an agreement that it relates to interpretability and transparency [3], [4]. In the context of HRI, explainability aims to improve the user’s understanding of the robot through clear and truthful explanations that align with the robot’s logic [5]. Explanations are also viewed as answers to “why” questions

Both authors are with Delft University of Technology, Faculty of Mechanical Engineering, Department of Cognitive Robotics, Delft, the Netherlands. E.M.Scheltinga@student.tudelft.nl, C.Pek@tudelft.nl

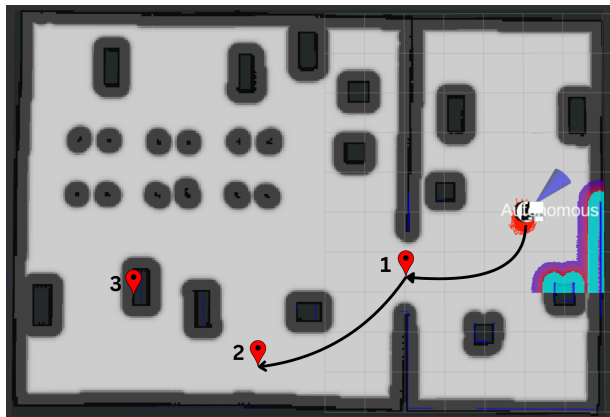


Fig. 1. Simple navigation task with three waypoints that our simulated TIAGo needs to reach. However, it fails to reach waypoint three, since there is an obstacle.

[5], and as an interaction between a human and a robot [6]. Most robots operate using the open-source Robot Operating System (ROS) framework that utilizes nodes to transmit data through topics. ROS also produces textual log messages in the *rosout* topic that give insights to the inner workings of the robot, which are comparable between different robots.

This study utilizes ROS log messages produced by the robot during navigation tasks to ground explanations, ensuring they reflect the robot’s logic. We will enhance the logs produced by ROS 1 by adding clearly defined custom log messages to increase the verbosity of the recorded log data.

To generate interactive and understandable explanations for users, we propose leveraging LLMs. These models are useful for interpreting extensive textual data such as ROS log messages due to their advanced natural language processing capabilities. This enables LLMs to provide understandable answers for questions about robot behavior [7], [8]. However, LLMs alone have limitations, such as hallucinations or providing answers in an undesired format [9]. To mitigate these limitations, we intend to adjust the LLMs weights by fine-tuning [10] these models on a dataset containing examples of question-answer pairs based on ROS log messages.

The purpose of the fine-tuned LLM is to interpret new ROS log messages and, based on this data, generate useful answers to users’ questions. Based on this approach, this work focuses on answering the following research question:

How can fine-tuning be used to generate useful explanations to improve explainability of autonomous robots’ behavior?

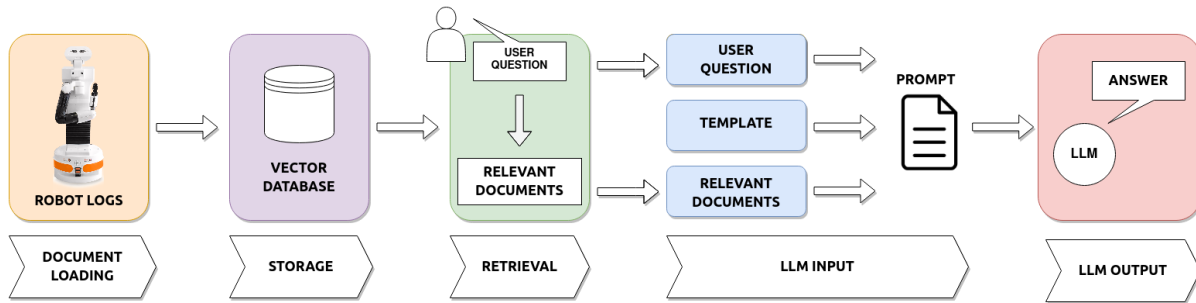


Fig. 2. Overview of the Retrieval Augmented Generation framework [11]

The structure of this paper is as follows: Section II presents a comprehensive review of related works. Section III details the method used to fine-tune the LLM for generating explanations using ROS log data. Section IV describes the implementation of this method. Section V displays the preliminary results collected so far.

II. RELATED WORK

This section briefly summarizes previous studies on robot explainability that leverage ROS log messages.

A study [12] uses an algorithm with if and else statements designed to read messages from specified topics and extract relevant information to generate simple natural language explanations for a navigation task. This method addresses two specific user questions, "What is the robot's current status?" and "Why have you changed the path?". For example, for the first question the algorithm compares the *previous_goal_id* with the *current_goal_id* and if these are different it returns "Navigation to a new goal has started". For the second question, if the distance from the goal location increases above a threshold and an obstacle is located within a threshold, the algorithm prints "I have changed the planned path because there was an obstacle". While it answers relevant questions for a navigation task, it limits user understanding to these questions and provides brief answers.

Another approach [8] employs LLMs to interpret ROS log messages, enhancing explainability. This approach involves splitting Rosbag files, containing ROS log messages, to fit the LLM's prompt size requirement and using basic prompt engineering to guide the LLM in generating relevant explanations. Specifically, this study utilizes single-shot prompting, where the LLM is provided with a single example to generate a similar response. The advantage of this approach is that by using LLMs, users can ask any question about the robot's behavior, instead of being limited to two predefined questions. Additionally, the responses to the questions contain greater depth and more context, which is provided by the LLMs natural language processing capabilities. However, the study does identify limitations concerning the quality and consistency of the log analysis.

Another study [11] introduces an explanation system using an LLM with Retrieval Augmented Generation (RAG) to interpret ROS log messages, as shown in Figure 2. Collected

ROS log data is stored in a vector database. Based on the provided user question, the RAG system retrieves relevant ROS log data to provide a broader context for LLM's response. The amount of provided contextual data is limited by the prompt size of the used LLM. The process involves forming a prompt for the LLM using the user question, a predefined prompt template, and the retrieved relevant ROS log data, which then produces the answer to the user question. The advantage of this method is that it retrieves relevant examples for the LLM, as an LLM aligns its response to the given context. Although, the amount of relevant context that can be provided and the quality of the retrieval system can limit the generated output.

Both studies [8], [11] build on [12], demonstrating the potential of combining LLMs with prompt engineering and RAG to interpret ROS log messages to provide comprehensible explanations to user questions. These studies highlight the potential of integrating LLMs for explanation generation based on ROS log data to enhance robot explainability. While current methods offer promising solutions for this problem, a new avenue that has yet to be investigated is fine-tuning. Fine-tuning involves adapting existing LLMs to specific tasks, in this case answering user questions based on provided ROS log messages, significantly improving the quality of the LLM responses. This method addresses the limitations of previous work, by eliminating constraints related to the maximum prompt size of an LLM and the quality of the retrieval system. Instead of relying on prompt engineering, the LLM is retrained on a comprehensive dataset that contains ROS log messages and corresponding question-answer pairs. This retraining process allows the LLM to better understand the context of the data for this specific task, leading to more accurate and relevant answers. Fine-tuning ensures that the LLM can leverage the entire dataset for generating responses, thereby providing richer and more precise explanations tailored to specific queries. Therefore, we will explore fine-tuning techniques to analyze ROS log data and generate answers to users' questions. I think

III. METHOD

The type of fine-tuning we intend to investigate is Parameter-Efficient Fine-Tuning (PEFT) to generate explanations based on the ROS log data. This section outlines the advantages of PEFT compared to traditional full fine-tuning.

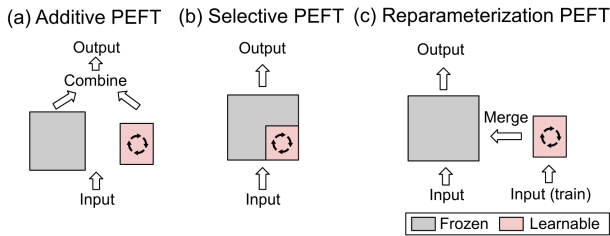


Fig. 3. Parameter-Efficient Fine-Tuning techniques [16]

The application of fine-tuning techniques to LLMs represents a significant advancement, enabling the customization of these models for a specific task [10], in this case question answering based on ROS log messages. Fine-tuning involves retraining an LLM using a dataset containing useful examples to optimize its performance for this task. There are various fine-tuning approaches. The most comprehensive fine-tuning approach is full fine-tuning, where the entire model is retrained, adjusting all its parameters to better suit the new dataset [13]. However, this approach is highly resource-intensive and time-consuming.

For ROS log interpretation tasks, a dataset of Rosbag files containing ROS log messages is required for fine-tuning. Given that such a dataset is not readily available online, it must be collected, limiting its volume. For this application, we propose using Parameter-Efficient Fine-Tuning (PEFT) as it involves selectively adjusting a small subset of the LLM’s parameters while leaving the majority unchanged [14], examples of PEFT techniques are shown in Figure 3. This targeted approach significantly reduces computational costs and the amount of fine-tuning data required compared to full fine-tuning, which retrains the entire model. The study by [15] indicates that PEFT outperforms traditional full fine-tuning, especially in limited-data scenarios, while offering benefits such as reduced memory and computational demands. Further research shows that increasing the model size enhances parameter-efficient fine-tuning outcomes more significantly than enlarging the pretraining or fine-tuning datasets [13], [14]. Employing large-scale LLMs with efficient fine-tuning practices like PEFT maximizes performance, making it highly relevant for tasks with constrained data and computational resources, such as those in ROS log interpretation tasks.

PEFT focuses on modifying critical parameters or layers within the model, such as the final layers, adapter modules, or specific attention heads, which are most relevant to the task at hand [14]. This method leverages the existing knowledge encoded in the pretrained model, allowing it to adapt to new tasks with minimal adjustments. By concentrating on key parameters, PEFT can achieve comparable performance to full fine-tuning while being both, computationally and data efficient [16], [17].

Low-Rank Adaptation (LoRA) [18] is a reparameterization PEFT technique that demonstrates potential in customizing LLMs for targeted tasks like automated code generation, while efficiently managing computational load [15].

LoRA optimizes rank decomposition matrices of dense layer changes during adaptation while keeping pre-trained weights frozen, achieving efficiency in storage and computation even with low rank [18]. In addition, a study [19] evaluates LoRA using datasets containing 1,000 examples to give an indication of the required size of a dataset. Therefore, we propose to utilize LoRA to fine-tune a LLM with a dataset of 900 examples to tailor it to answering user questions based on a robot’s ROS log messages.

IV. EXPERIMENTAL SETUP

The goal of the proposed experiments is to evaluate the performance of PEFT and the quality of the explanations it generates. This section details the collection, pre-processing, and utilization of fine-tuning data for PEFT. Additionally, the testing conditions and evaluation methods used to assess the effectiveness of PEFT in generating useful explanations are discussed.

A. Collecting ROS log data

We collect Rosbag files containing ROS log messages with TIAGO in simulation using the standard software stack provided by PAL Robotics, that is tasked to move along waypoints:

- **Scenario 1:** with known obstacles, and all paths are feasible.
- **Scenario 2:** with unknown obstacles in its path, yet all paths are feasible.
- **Scenario 3:** with known obstacles, yet not all paths are feasible. As shown in Figure 1.
- **Scenario 4:** with unknown obstacles in its path, and not all paths are feasible.
- **Scenario 5:** with no unknown obstacles, but at least one of the paths goes out of bounds.

In these scenarios, known obstacles refer to obstacles that are present in the costmap the robot uses to plan its paths and unknown obstacles are not. For each scenario, small variations are induced, such as in the number of waypoints and waypoint positions, resulting in the collection of a total of 75 Rosbag files containing the ROS log messages produced by the */rosout* topic. The Rosbag starts recording simultaneously with the initialization of the navigation task to minimize the presence of irrelevant ROS log messages.

B. Pre-processing data

We label each Rosbag file with multiple questions and corresponding answers, resulting in 900 data instances containing a Rosbag file, a question and an answer. The questions used in our research are a combination of those from the work by Sobrín-Hidalgo et al. [11] and additional ones we propose, shown in Appendix I.

These questions are included in the fine-tuning data, and the answers for each of these questions related to a Rosbag file are generated using OpenAI’s GPT-4o. For 900 answers, we deemed it more practical to generate these using GPT-4o instead of manually, as this is time-consuming and LLMs such as GPT-4o have excellent capabilities to generate useful

explanations with the right prompt. The prompt used for this data generation is shown in Appendix II employs the prompting principles detailed in [20]. Together, these 900 instances containing a Rosbag file, a question and an answer make up the fine-tuning data.

C. Parameter-Efficient Fine-Tuning

The Parameter-Efficient Fine-Tuning is carried out using the collected dataset composed of ROS logs, user questions and corresponding answers on two LLMs: GPT-3.5 and an open-source model Mistral 8x7B. These two models were selected due to their size and performance.

D. Testing

We conduct four different tests using the two fine-tuned LLMs and the two original LLMs. Each test will be evaluated using an existing Rosbag file from a previous study by Sobrín-Hidalgo et al. [11], and Rosbag files collected from the physical TIAGo robot, both performing navigation tasks.

E. Evaluation

We evaluate the fine-tuned LLM using a combination of objective metrics and a subjective questionnaire. The set of metrics aims to quantify the system’s performance, and a questionnaire assesses the quality of the generated explanations.

The proposed metrics are adapted from [11] for a PEFT application:

- **M1:** Number of messages published in the Rosbag file.
- **M2:** Answer generation time (s)

M1 gives an indication of the size and complexity of the Rosbag file that is being interpreted. M2 refers to the time taken to generate an explanation for the fine-tuned model. The hardware we will use to conduct the final experiments is still to be decided.

Additionally, we will use a questionnaire developed by [11] to assess the quality of the explanations generated by the fine-tuned LLM. This questionnaire, targeted towards roboticists, evaluates the explanations based on six criteria adapted from the same study given the ROS log messages. Roboticists with prior knowledge of ROS log messages are best suited to make this evaluation because they possess the necessary understanding to provide a well-grounded assessment. These criteria, as shown in Table II in Appendix III, focus on user expectation and satisfaction and are scored using a 5-point Likert scale.

This evaluation aims to validate the potential of the proposed method by providing an assessment of the quality and reliability of the produced explanations. Furthermore, it can be used to compare with the evaluation of the RAG method proposed by [11].

V. PRELIMINARY RESULTS

This section presents preliminary results of this study, showcasing the responses provided by GPT-3.5-turbo before and after it is fully fine-tuned using the OpenAI API. The aim of these results is to show the improvement in responses

through fine-tuning LLMs using the collected data. The next step in the research is to implement and test PEFT.

The preliminary results show the response of the LLM based on new ROS log messages recorded during a navigation task of scenario 3 with 3 waypoints, as shown in Figure 1. A simplified version of the ROS log messages is given below:

```
Navigating to waypoint 1...
Reached waypoint 1
Navigating to waypoint 2...
Reached waypoint 2
Navigating to waypoint 3...
Failed to get a plan.
Rotate recovery behavior started.
Aborting because a valid plan could
not be found. Even after
executing all recovery
behaviors.
Failed to reach waypoint 3
```

The full ROS log messages recorded in the Rosbag are shown in Appendix IV.

An overview of the answers generated by GPT-3.5 before and after fine-tuning is shown in Table III and Table IV respectively in Appendix V. These results show how fine-tuning using the collected data has improved the explanations provided by the LLM. The fine-tuned LLM provides answers in an improved format, as shown in the responses to question 1, 2 and 10. Additionally, it provides clearer references to the log messages provided to ground its explanations, demonstrated in the responses to questions 4 to 7. The combination of these improved qualities show the potential of fine-tuning and we look forward to continuing this research by testing PEFT.

VI. CONCLUSIONS

In this research, we explore the potential of PEFT to interpret textual ROS log messages and generate useful explanations for robot behaviors. Our preliminary results demonstrate that fine-tuning improves the response of an LLM to user questions about a simple navigation task, demonstrating promising potential for improving robot explainability in HRI. We look forward to comparing these results to the answers generated by LLMs that are fine-tuned using PEFT.

We anticipate presenting final results that highlight the potential of PEFT of LLMs for generating insightful explanations based on ROS log data. Furthermore, through this research, we hope to contribute to the development of explainability systems that can be integrated with various ROS-operating robots. Such systems are crucial for building trust with human users and addressing robot failures effectively.

Overall, our work underscores the potential of PEFT in advancing robot explainability, and we look forward to sharing further insights as our study progresses.

REFERENCES

- [1] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," vol. 6, pp. 52138–52160. [Online]. Available: <https://ieeexplore.ieee.org/document/8466590/>
- [2] T. Sakai and T. Nagai, "Explainable autonomous robots: a survey and perspective," vol. 36, no. 5, pp. 219–238. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/01691864.2022.2029720>
- [3] A. Rosenfeld, "Better metrics for evaluating explainable artificial intelligence: Blue sky ideas track."
- [4] G. Vilone and L. Longo, "Explainable artificial intelligence: a systematic review," publisher: [object Object] Version Number: 4. [Online]. Available: <https://arxiv.org/abs/2006.00093>
- [5] A. Rosenfeld and A. Richardson, "Explainability in human-agent systems," vol. 33, no. 6, pp. 673–705. [Online]. Available: <http://link.springer.com/10.1007/s10458-019-09408-y>
- [6] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance," vol. 5, publisher: Frontiers. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fcomp.2023.1096257>
- [7] L. Fernández-Becerra, M. A. González-Santamarta, A. M. Guerrero-Higuera, F. J. Rodríguez-Lera, and V. M. Olivera, "Enhancing trust in autonomous agents: An architecture for accountability and explainability through blockchain and large language models." [Online]. Available: <http://arxiv.org/abs/2403.09567>
- [8] M. A. González-Santamarta, L. Fernández-Becerra, D. Sobrín-Hidalgo, A. M. Guerrero-Higuera, I. González, and F. J. R. Lera, "Using large language models for interpreting autonomous robots behaviors," publisher: [object Object] Version Number: 1. [Online]. Available: <https://arxiv.org/abs/2304.14844>
- [9] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models." [Online]. Available: <https://arxiv.org/abs/2401.11817>
- [10] M. R. J. K. VM, H. Warriar, and Y. Gupta, "Fine tuning LLM for enterprise: Practical guidelines and recommendations." [Online]. Available: <http://arxiv.org/abs/2404.10779>
- [11] D. Sobrín-Hidalgo, M. A. González-Santamarta, A. M. Guerrero-Higuera, F. J. Rodríguez-Lera, and V. Matellán-Olivera, "Explaining autonomy: Enhancing human-robot interaction through explanation generation with large language models," publisher: [object Object] Version Number: 1. [Online]. Available: <https://arxiv.org/abs/2402.04206>
- [12] L. Fernández-Becerra, M. A. González-Santamarta, D. Sobrín-Hidalgo, A. M. Guerrero-Higuera, F. J. R. Lera, and V. M. Olivera, "Accountability and explainability in robotics: A proof of concept for ROS 2- and nav2-based mobile robots," in *International Joint Conference 16th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2023) 14th International Conference on European Transnational Education (ICEUTE 2023)*, ser. Lecture Notes in Networks and Systems. Springer Nature Switzerland, pp. 3–13.
- [13] B. Zhang, Z. Liu, C. Cherry, and O. Firat, "When scaling meets LLM finetuning: The effect of data, model and finetuning method." [Online]. Available: <https://openreview.net/forum?id=5HCnKDeTws>
- [14] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H.-T. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, and M. Sun, "Parameter-efficient fine-tuning of large-scale pre-trained language models," vol. 5, no. 3, pp. 220–235, publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s42256-023-00626-4>
- [15] M. Weyssow, X. Zhou, K. Kim, D. Lo, and H. Sahraoui, "Exploring parameter-efficient fine-tuning techniques for code generation with large language models," publication Title: arXiv e-prints ADS Bibcode: 2023arXiv230810462W. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2023arXiv230810462W>
- [16] Z. Han, C. Gao, J. Liu, Jeff, Zhang, and S. Qian Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," publication Title: arXiv e-prints ADS Bibcode: 2024arXiv240314608H. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2024arXiv240314608H>
- [17] Z. Fu, H. Yang, A. Man-Cho So, W. Lam, L. Bing, and N. Collier, "On the effectiveness of parameter-efficient fine-tuning," publication Title: arXiv e-prints ADS Bibcode: 2022arXiv221115583F. [Online]. Available: <https://ui.adsabs.harvard.edu/abs/2022arXiv221115583F>
- [18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models." [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>
- [19] J. Wang, G. Yang, W. Chen, H. Yi, X. Wu, and Q. Lao, *MLAE: Masked LoRA Experts for Parameter-Efficient Fine-Tuning*.
- [20] S. M. Bsharat, A. Myrzakhan, and Z. Shen, "Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4." [Online]. Available: <http://arxiv.org/abs/2312.16171>

APPENDIX

APPENDIX I

TABLE I
QUESTIONS FOR EACH SCENARIO

Questions for all scenarios:

Q1: How many waypoints were received?*

Q2: Which waypoints were received?*

Q3: What were the coordinates of all the waypoints the robot received?

Q4: Were all the received waypoints successfully reached?*

Q5: What was the first waypoint the robot reached?

Q6: Where was the last waypoint the robot reached?

Q7: Were all paths feasible during the navigation task?

Q8: Were there any waypoints the robot failed to reach?

Q9: Were there any obstacles the robot encountered during the navigation task?

Q10: What is the task the robot had to perform?*

Additional questions for scenario 2 & 4:

Q11: Where was the encountered obstacle located?

Q12: How did the robot respond to detected obstacles?

Additional questions for scenario 3 & 4 & 5:

Q13: Why were one or more paths not feasible?

Q14: How did the robot deal with unfeasible path(s)?

The questions indicated with a * are from the work by Sobrín-Hidalgo et al. [11].

APPENDIX II PROMPT USED FOR ANSWER GENERATION

```
{"role": "system", "content": "###Instruction### You are an explainability AI specialised in analyzing and interpreting ROS log messages for autonomous robots. Your task is to provide clear, concise and factual explanations based on the logs generated during robot operations. Focus on delivering precise answers to the user's questions."}
{"role": "user", "content": "###Context### Based on the ROS log messages in the following document: [insert ROS log messages]. ###Question### [insert question]"}}
```

This prompt was designed using the prompting principles detailed in [20].

APPENDIX III ASSESSMENT CRITERIA

TABLE II
CRITERIA FOR EXPLANATION QUALITY AND SATISFACTION

Focus	Criteria
Quality	I consider the explanation to be unnecessarily long
	The explanation is written in clear and understandable language
	I believe that the explanation is consistent with the robot's behavior shown in the logs
Satisfaction	The explanation has improved my understanding of what happened during the robot task
	I consider that the explanation evidences the behavior of the robot
	I consider that the explanation is sufficiently reliable and accurate to understand the behavior

APPENDIX IV
FULL ROS LOG MESSAGE

```
Started controllers: hand_controller
Controller Spawner: Loaded controllers: arm_velocity_trajectory_controller
"Started controllers: joint_state_controller, torso_controller, head_controller,
  arm_controller, mobile_base_controller"
Started controllers: force_torque_sensor_controller
Started controllers: imu_sensor_controller
Waiting for move_base action server...
Connected to move_base action server.
"Sending goal 1 to move_base: x=1.5, y=-3.0"
Navigating to waypoint 1...
"Transitioning to ACTIVE (from WAITING_FOR_GOAL_ACK, goal:
  /tiago_autonomous_navigation-1-25.149)"
GOAL Reached!
"Transitioning to WAITING_FOR_RESULT (from ACTIVE, goal:
  /tiago_autonomous_navigation-1-25.149)"
"Transitioning to DONE (from WAITING_FOR_RESULT, goal:
  /tiago_autonomous_navigation-1-25.149)"
Reached waypoint 1
"Sending goal 2 to move_base: x=4.0, y=-7.0"
Navigating to waypoint 2...
"Transitioning to ACTIVE (from WAITING_FOR_GOAL_ACK, goal:
  /tiago_autonomous_navigation-2-34.359)"
GOAL Reached!
"Transitioning to WAITING_FOR_RESULT (from ACTIVE, goal:
  /tiago_autonomous_navigation-2-34.359)"
"Transitioning to DONE (from WAITING_FOR_RESULT, goal:
  /tiago_autonomous_navigation-2-34.359)"
Reached waypoint 2
"Sending goal 3 to move_base: x=2.0, y=-10.0"
Navigating to waypoint 3...
"Transitioning to ACTIVE (from WAITING_FOR_GOAL_ACK, goal:
  /tiago_autonomous_navigation-3-46.368)"
Failed to get a plan.
Clearing both costmaps outside a square (3.00m) large centered on the robot.
Clearing both costmaps outside a square (1.50m) large centered on the robot.
Rotate recovery behavior started.
Failed to get a plan.
Aborting because a valid plan could not be found. Even after executing all recovery
  behaviors
"Transitioning to WAITING_FOR_RESULT (from ACTIVE, goal:
  /tiago_autonomous_navigation-3-46.368)"
"Transitioning to DONE (from WAITING_FOR_RESULT, goal:
  /tiago_autonomous_navigation-3-46.368)"
Failed to reach waypoint 3
```

APPENDIX V
PRELIMINARY RESULTS

TABLE III
ANSWERS GENERATED BY GPT-3.5-TURBO-0125 (NOT FINE-TUNED)

Question	Answer
Q1. How many waypoints were received?	Three waypoints were received based on the log messages.
Q2. Which waypoints were received?	Waypoints 1 and 2 were successfully reached based on the log messages. Waypoint 3 was not reached due to a failed plan.
Q3. What were the coordinates of all the waypoints the robot received?	The coordinates of the waypoints the robot received were: - Waypoint 1: $x=1.5, y=-3.0$ - Waypoint 2: $x=4.0, y=-7.0$ - Waypoint 3: $x=2.0, y=-10.0$
Q4. Were all waypoints successfully reached?	No, not all the waypoints were successfully reached. Waypoints 1 and 2 were successfully reached, but waypoint 3 was not reached due to a failed plan.
Q5. What was the first waypoint the robot reached?	The first waypoint the robot reached was waypoint 1, with coordinates $x=1.5$ and $y=-3.0$.
Q6. Where was the last waypoint the robot reached?	The last waypoint the robot reached was waypoint 2, with coordinates $x=4.0$ and $y=-7.0$. Waypoint 3 was not reached due to a failed plan.
Q7. Were all paths feasible during the navigation task?	No, not all paths were feasible during the navigation task. While the robot successfully reached waypoints 1 and 2, it failed to reach waypoint 3 due to a failed plan.
Q8. Were there any waypoints the robot failed to reach?	Yes, there was one waypoint that the robot failed to reach. The robot successfully reached waypoints 1 and 2, but it failed to reach waypoint 3 due to a failed plan.
Q9. Were there any obstacles the robot encountered during the navigation task?	Based on the provided ROS log messages, it appears that the robot encountered obstacles or other issues that prevented it from successfully reaching waypoint 3. The log indicates that the robot failed to get a plan and then executed recovery behaviors, such as clearing costmaps and initiating a rotate recovery behavior, but was ultimately unable to navigate to waypoint 3. This suggests that obstacles or navigation difficulties impeded the robot's progress.
Q10. What is the task the robot had to perform?	The task the robot had to perform based on the provided ROS log messages was autonomous navigation to a series of waypoints. The robot received three goals to navigate to specific coordinates (waypoint 1 at $x=1.5, y=-3.0$, waypoint 2 at $x=4.0, y=-7.0$, and waypoint 3 at $x=2.0, y=-10.0$). The robot successfully reached waypoints 1 and 2 but encountered difficulties and failed to reach waypoint 3 due to the inability to generate a valid plan despite executing recovery behaviors.

TABLE IV
ANSWERS GENERATED BY GPT-3.5-TURBO-0125 (FULLY FINE-TUNED)

Question	Answer
Q1. How many waypoints were received?	Based on the provided ROS log messages, a total of three waypoints were received. The goals were: <ol style="list-style-type: none"> 1. Waypoint 1: $x=1.5, y=-3.0$ 2. Waypoint 2: $x=4.0, y=-7.0$ 3. Waypoint 3: $x=2.0, y=-10.0$
Q2. Which waypoints were received?	According to the provided ROS log messages, the following waypoints were received: <ol style="list-style-type: none"> 1. Waypoint 1: $x=1.5, y=-3.0$ 2. Waypoint 2: $x=4.0, y=-7.0$ 3. Waypoint 3: $x=2.0, y=-10.0$ <p>Although the robot failed to reach waypoint 3, the goal for it was still sent and received by the system.</p>
Q3. What were the coordinates of all the waypoints the robot received?	The robot received the following coordinates for the waypoints in the provided ROS log messages: <ol style="list-style-type: none"> 1. Waypoint 1: $x=1.5, y=-3.0$ 2. Waypoint 2: $x=4.0, y=-7.0$ 3. Waypoint 3: $x=2.0, y=-10.0$
Q4. Were all waypoints successfully reached?	No, not all the waypoints were successfully reached. Waypoints 1 and 2 were successfully reached based on the log messages indicating "GOAL Reached!" and the subsequent transitions to "DONE". However, waypoint 3 was not reached as indicated by the failure to generate a valid plan and remaining in the "DONE" state without reaching the goal.
Q5. What was the first waypoint the robot reached?	The first waypoint the robot reached was at coordinates $x=1.5, y=-3.0$. This is indicated in the log messages by the successful transition from "WAITING_FOR_GOAL_ACK" to "ACTIVE," followed by "GOAL Reached!" and ultimately "DONE" for the goal with ID '/tiago_autonomous_navigation-1-25.149'.
Q6. Where was the last waypoint the robot reached?	The last waypoint the robot successfully reached was at coordinates $x = 4.0$ and $y = -7.0$, which is waypoint 2. This is confirmed by the transition log messages indicating that the robot successfully navigated to this waypoint and then completed all relevant state transitions. The subsequent attempt to reach waypoint 3 was unsuccessful, as detailed in the recovery attempts and final failure message.
Q7. Were all paths feasible during the navigation task?	No, not all planned paths were feasible during the navigation task. The robot successfully reached waypoints 1 and 2, but it failed to reach waypoint 3. The logs indicate that the system failed to generate a valid plan despite multiple attempts and recovery behaviors, resulting in the aborting of the mission to reach waypoint 3.
Q8. Were there any waypoints the robot failed to reach?	Yes, the robot failed to reach waypoint 3. The log messages indicate that the robot encountered issues during the planning process, and all recovery behaviors failed to generate a valid plan. This led to the abortion of the mission to reach waypoint 3.
Q9. Were there any obstacles the robot encountered during the navigation task?	The ROS log messages do not explicitly state that the robot encountered specific obstacles during the navigation task. However, the fact that the robot failed to reach waypoint 3 and the recovery behaviors (such as clearing costmaps and performing rotate recovery) were unsuccessful in getting a valid plan suggests that there may have been obstacles in the environment preventing the robot from finding a viable path to waypoint 3. The repetitive nature of the clearing attempts and ultimate failure to re-plan indicate a potential obstruction rather than simple pathfinding issues.
Q10. What is the task the robot had to perform?	The task the robot had to perform was a sequence of navigation actions. Specifically, the log messages indicate that the robot was given three consecutive goals (waypoints) to reach via the "move_base" action server. <ol style="list-style-type: none"> 1. The first goal was to reach the coordinates $x = 1.5, y = -3.0$. 2. The second goal was to reach the coordinates $x = 4.0, y = -7.0$. 3. The third goal was to reach the coordinates $x = 2.0, y = -10.0$. <p>The robot successfully navigated to and reached the first two waypoints, but failed to reach the third waypoint due to an inability to generate a valid plan despite executing recovery behaviors.</p>