Are You Being Tracked? Discover the Power of Zero-Shot Trajectory Tracing with LLMs!

Huanqi Yang[1], Sijie Ji[2], Rucheng Wu[1], Weitao Xu[1]

[1]City University of Hong Kong

[2]University of Hong Kong

Abstract—There is a burgeoning discussion around the capabilities of Large Language Models (LLMs) in acting as fundamental components that can be seamlessly incorporated into Artificial Intelligence of Things (AIoT) to interpret complex trajectories. This study introduces LLMTrack, a model that illustrates how LLMs can be leveraged for Zero-Shot Trajectory Recognition by employing a novel single-prompt technique that combines role-play and think step-by-step methodologies with unprocessed Inertial Measurement Unit (IMU) data. We evaluate the model using real-world datasets designed to challenge it with distinct trajectories characterized by indoor and outdoor scenarios. In both test scenarios, LLMTrack not only meets but exceeds the performance benchmarks set by traditional machine learning approaches and even contemporary state-of-the-art deep learning models, all without the requirement of training on specialized datasets. The results of our research suggest that, with strategically designed prompts, LLMs can tap into their extensive knowledge base and are well-equipped to analyze raw sensor data with remarkable effectiveness.

Index Terms—Large Language Models, AIoT, Tracking, Trajectory

I. INTRODUCTION

Recent advancements in Large Language Models (LLMs) have showcased remarkable capabilities [1]–[3], particularly evident in instances such as Sora [4] and Claude-3 [5]. Sora exemplifies how LLMs can intuitively grasp a wide range of world phenomena, from the principles of physics to the complexities of societal and artistic concepts, simulating a child's learning process. These LLMs not only excel in a diverse array of downstream tasks but also demonstrate high-level human-like reasoning and exceptional generalization abilities. The potential of LLMs extends to the realm of the Artificial Intelligence of Things (AIoT), where such models can be utilized to enhance the intelligence of interconnected devices, enabling them to interact with and understand the world in more sophisticated and refined ways.

Despite these notable achievements, certain scholars argue that existing LLMs have merely a superficial grasp of knowledge and remain a considerable distance from attaining a profound comprehension of the physical realm [6]. This shortcoming is often attributed to their reliance on training using extensive collections of textual and visual content from the internet, which may result in less-than-optimal performance when evaluating digital and temporal data sequences. Moreover, these models often face challenges in meaningfully interacting with the real world, such as when they are assigned to generate precise and efficient control strategies.
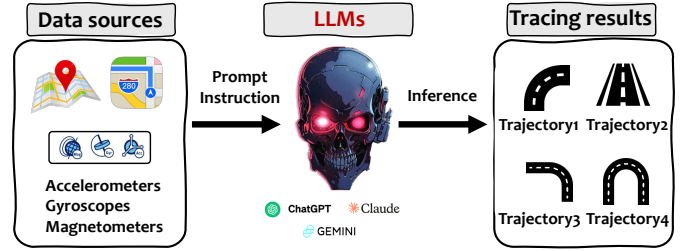

Figure 1: Workflow of LLMTrack.

The lively discussion around this topic has fueled a growing interest in examining how LLMs could become core components that are intricately interwoven with Cyber-Physical Systems (CPS) to decode aspects of the physical environment. Specifically, some researchers have introduced the notion of HARGPT [7], which investigates the potential of employing LLMs for human activity recognition, yielding promising results. These findings suggest a valuable application of LLMs in understanding and interpreting human behaviors through data analysis, indicating a significant advancement in the field. However, the full scope of what LLMs might achieve in terms of modeling and understanding the physical world in other tasks has not been thoroughly investigated.

Inspired by the strides made with LLMs and the prospects of HARGPT, this paper sets out by focusing on the fundamental application of trajectory tracing within Cyber-Physical Systems. It seeks to investigate and understand the existing abilities of LLMs by inputting unprocessed sensor data coupled with a basic prompt. As Shown in Fig. 1, the unprocessed IoT sensor data is input into several renowned LLMs including ChatGPT, Claude, and Gemini, resulting in the models producing trajectory tracking results.

The performance evaluation spans two different environments: indoor and outdoor, with each having four unique types of trajectories. The benchmarks set for comparison include both traditional machine learning models and contemporary state-of-the-art deep learning models. The experimental findings reveal that LLMs possess the capability to execute zero-shot tracking with raw sensor data effectively. When compared to traditional methods, not only do LLMs surpass them, but they also attain an average accuracy above 80%. It is particularly noteworthy that, in contrast to learning-based models which tend to suffer from performance drops when dealing with novel data and usually require retraining or fine-tuning for
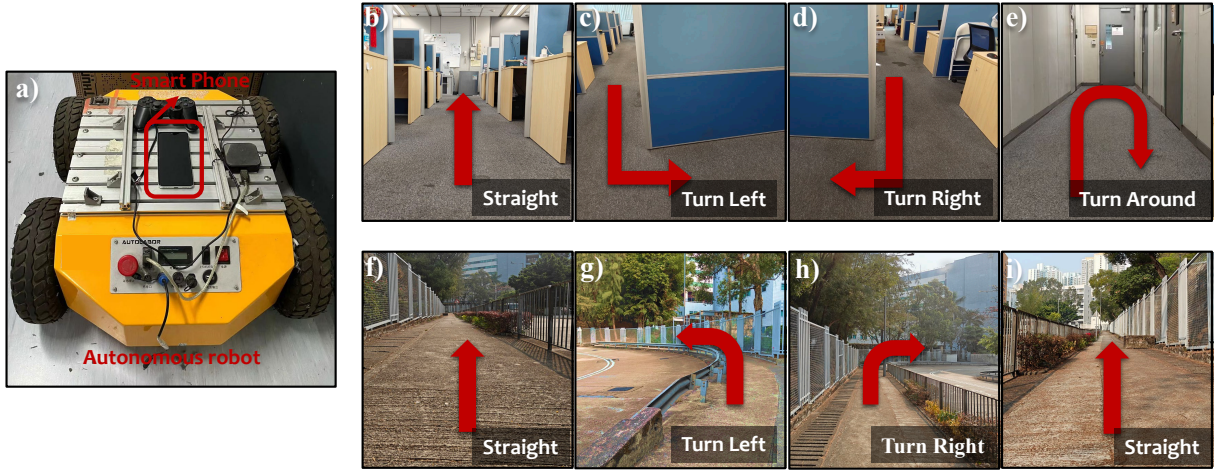
Figure 2: Experimental settings. a) Experiment devices. b)–i) Eight different scenes in indoor and outdoor environments.

particular datasets, LLMs demonstrate considerable resilience. We introduce LLMTrack and highlight the following points:

- LLMTrack uncovers the ability of LLMs to act as zero-shot Trajectory Tracers without the necessity for any fine-tuning or the use of prompts crafted with domain-specific knowledge.
- LLMTrack affirms the adeptness of LLMs in interpreting IoT sensor data and executing tasks within the physical realm.

Moreover, we engage in an in-depth analysis and discussion about the insights and significant findings from employing LLMs with IoT sensor data, which sets the stage for further studies on the integration of LLMs into AIoT systems.

II. LLMTRACK: ZERO-SHOT TRACKING WITH LLMS

A. Experiments

To examine the proficiency of LLMs in identifying robotic trajectories from raw Inertial Measurement Unit (IMU) data, we structured our investigation into two experimental stages, each tailored to different scenarios. The preliminary stage focused on ascertaining the capability of LLMs to recognize trajectories in an indoor setting, characterized by uncomplicated navigational paths. Moving to the second stage, the aim was to determine the extent to which these models could discern between trajectories within outdoor environments, which are not only complex due to the irregular ground surfaces but also notoriously difficult to distinguish, presenting a substantial challenge.

1) Dataset Setup: To support these experiments, we employed two distinct datasets, one captured indoors and the other outdoors, both collected by an autonomous robot equipped with a smartphone, as depicted in Fig. 2. Additional information is detailed in the following.

Indoor setting: The indoor dataset encompasses a diverse array of robotic trajectories traversing inside a building, meticulously recorded via an IMU-equipped smartphone. For every trajectory, we harnessed the 9-axis IMU dataset, which is annotated with explicit labels: 'straight', 'turn right', 'turn left', and 'turn around'. An illustration of the sample raw data is showcased in Fig. 3a to Fig. 3c. This IMU consistently operates at a default sampling rate of 100Hz. To benchmark the performance of LLMs against baseline models that require training data, we divided our dataset into training, validation, and seen test and unseen test sets at a ratio of 3:1:1:1. Furthermore, considering the necessity to feed raw IMU data as tokens into LLMs, we performed a downsampling of the IMU data to a frequency of 3Hz specifically for LLM inputs.

Outdoor setting: The outdoor dataset comprises various robot trajectories collected around a campus environment, utilizing 9-axis IMU data from smartphones mounted on top of the robots. We chose four trajectories as shown in Fig.2. An illustration of the sample raw data is showcased in Fig. 3d to Fig. 3f. Similar to the partitioning approach used for the indoor dataset, the outdoor dataset is also segmented into distinct sets for training, validation, and testing (both seen and unseen) at a ratio of 3:1:1:1. To accommodate the input requirements of LLMs, the IMU data is further downsampled to 3Hz before being tokenized for model processing.

2) Baseline: As benchmarks for comparison, we employ traditional machine learning models as well as deep learning models. These models require a training phase to develop their capacity for classification.

Random Forest [8]: Random forests (RF) is an ensemble learning technique used for classification tasks. It builds numerous decision trees during the training process and is known for its ability to effectively manage classes of varying sizes within classification problems. For our implementation, we utilize the Random Forest model from the scikit-learn library, applying the default parameters provided.

Support Vector Machine [9]: The support vector machine (SVM) is a supervised learning model designed for both classification and regression tasks. It is known for its capacity to find the maximum margin in a dataset, often resulting in robust classification boundaries. Our experiments leverage

(a) Indoor turn around accelerations   (b) Indoor turn around gyroscopes   (c) Indoor turn around magnetometers

(d) Outdoor turn right accelerations   (e) Outdoor turn right gyroscopes   (f) Outdoor turn right magnetometers
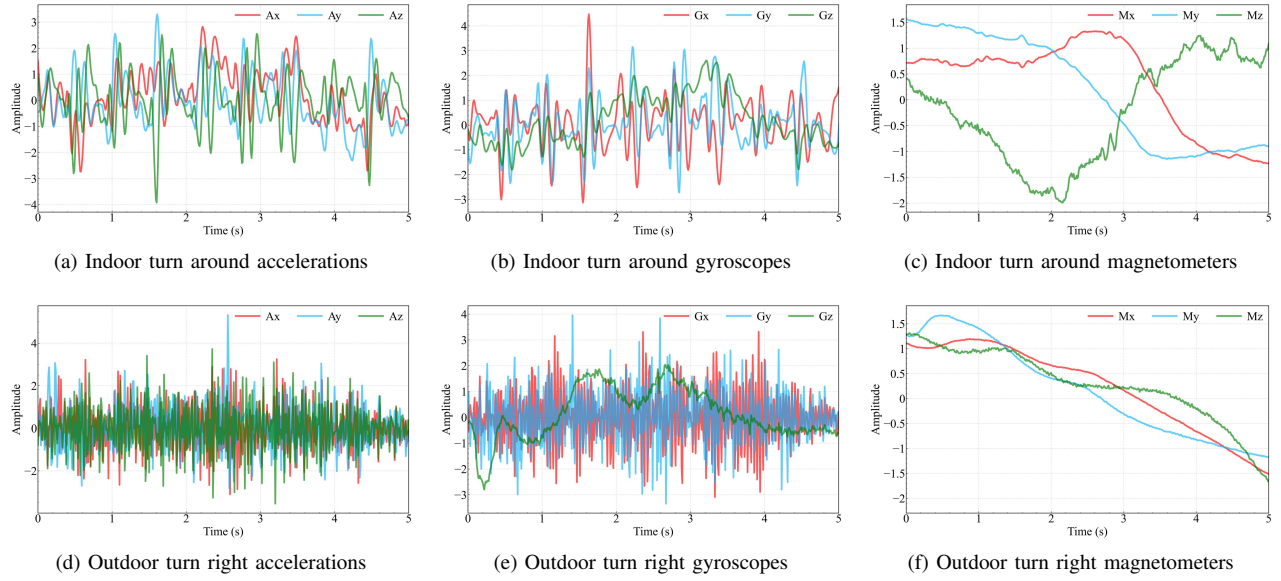
Figure 3: IMU data visualization of two scenes including indoor turn right and outdoor turn right.

the SVM implementation from scikit-learn, specifically using Gaussian kernels.

CNN [10]: The Convolutional Neural Network is a deep learning architecture that excels in automatically extracting features from multichannel time series signals, such as those gathered by sensors on a robot. CNN is adept at learning intricate features from trajectory data and often outperform traditional approaches in classification accuracy.

LSTM [11]: Long Short-Term Memory networks (LSTMs) are a specialized type of model designed explicitly for tasks requiring the understanding and memory of temporal sequences. Standing out from other foundational models, LSTMs handle input features in a sequential manner through time. This sequential processing capability allows LSTMs to excel in applications where the temporal dimension is critical, such as speech recognition [12], channel forecasting [13], and gesture recognition [14].

3) Prompt Structure: LLMs have been recognized for their proficiency in few-shot learning, wherein they utilize prompts structured text or templates—to steer their output towards specific tasks [15]. For instance, Nori et al. reported leveraging instructional cues to tap into the extensive medical expertise embedded within LLMs, setting new benchmarks across various datasets [16]. However, the limitation of this prompt-based approach is the requirement of crafting tailored question-and-answer frameworks, which can confine LLMs to narrowly-defined tasks [17]. Additionally, it has been observed that such tailor-made prompts can disrupt the natural inferencing process known as the Chain-of-Thought (CoT), potentially degrading performance [18]. To address these limitations, our strategy involves constructing prompts that are fundamentally simple, employing role-playing and incremental thought processes to naturally elicit the inherent capabilities of LLMs without depending on predefined answer frameworks.

Figure 4: Chain-of-thought prompt design for LLMTrack.

The construction of our prompt, as visualized in Fig. 4, is elegantly minimalist, comprising solely of an initial instruction followed by a probing question. The directive is designed to activate the LLM's reservoir of specialized knowledge concerning IMU technology. Simultaneously, the question furnishes precise context about the data acquisition process, the downsampled sequence of raw IMU data, and the plausible categories of trajectories being investigated. Our prompt culminates with an invitation for the LLM to "I would appreciate a step-by-step analysis of your reasoning process" a tactic intended to provoke a comprehensive CoT progression within the LLM. This method is championed by existing research [15], [19], for its efficacy in refining the precision of LLM outputs. As illustrated in Fig. 5, by foregoing a restricted response

ChatGPT

Certainly, let's analyze the provided IMU data step by step to deduce the specific trajectory performed by the robot. We'll look at the accelerometer, gyroscope, and magnetometer data separately and then synthesize our findings.

1. Overall Analysis

2. Expert Knowledge

Accelerometer Analysis:
The accelerometer measures linear acceleration. Here's what the data suggests:
- **x-axis**: The values fluctuate slightly around zero, indicating no persistent motion in the lateral direction relative to the robot.
- **y-axis**: There's a noticeable decrease, reaching a minimum of -0.7, which may indicate a forward deceleration or a backward acceleration at some point.
- **z-axis**: The values show minor fluctuations around 0.1, which could be due to noise or the robot's vertical motion, such as going over a bump.

Gyroscope Analysis:
The gyroscope measures angular velocity. Here's what the data suggests:
- **x-axis**: Constant zero values indicate no rotation around the robot's longitudinal axis.
- **y-axis**: Also constant zero values, indicating no rotation around the lateral axis.
- **z-axis**: Starts at zero, then shows a change, peaking at 0.8 and returning to zero, indicating a rotation around the vertical axis.

Magnetometer Analysis:
The magnetometer measures magnetic fields and can indicate orientation changes. Here's what the data suggests:
- **x-axis**: There is a significant change from 24.2 to -13.3, which suggests a substantial change in the robot's orientation relative to the Earth's magnetic field.
- **y-axis**: The change from -1.6 to -7.6 is relatively moderate.
- **z-axis**: The values decrease steadily, which is consistent with a change in orientation but not as indicative as the x-axis changes.

3. Professional Analysis

Synthesis and Conclusion:
- **Accelerometer**: The data does not show significant forward or lateral acceleration, suggesting the robot was not moving fast or changing speeds dramatically.
- **Gyroscope**: The z-axis data shows a clear rotation at one point during the sequence, which points to a turn being made.
- **Magnetometer**: The large change in the x-axis and the steady decrease in the z-axis support the idea of a turn.

Now, to classify the type of trajectory:
- **Straight**: With a straight path, we would expect minimal changes in all IMU sensors, except for a constant forward acceleration if starting from rest or deceleration if coming to a stop.
- **Turn Left**: A left turn would be characterized by a positive angular velocity around the z-axis (assuming a conventional coordinate system where x is forward, y is left, and z is up).

Turn Around: A turn around would involve a more significant and sustained angular velocity around the z-axis, possibly more than 180 degrees.

Given the gyroscope data (z-axis goes from -0.0 to 0.8 and back to 0.0), there was a significant turn. The magnetometer data supports this with a large swing in the x-axis (from 24.2 to -13.3), indicating a substantial change in orientation.
Since the accelerometer does not show significant forward acceleration, and considering the gyroscope and magnetometer data, it is likely that the robot performed a **turn left** or **turn around**. The gyroscope z-axis data indicate the turn was significant but not sustained over a long period (it went back to 0.0), and thus, it is more indicative of a **turn left** rather than a full 360-degree **turn around**.
Therefore, based on the provided data and these analyses, the specific trajectory performed by the robot is most likely a **turn left** maneuver.
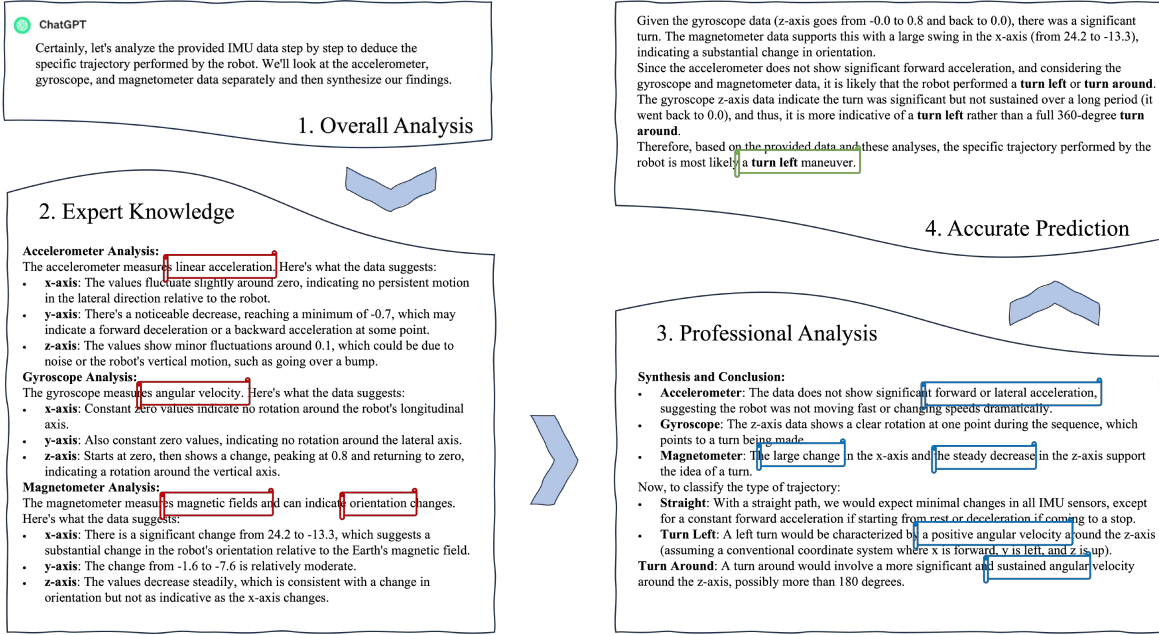
4. Accurate Prediction

Figure 5: Detailed step-by-step inference generated by GPT4 with a turn-left example.

Table I: Overall performance. (● for chosen, ○ for unchosen, DO*: direct output; CoT*: chain-of-thought.)

Models	Scenarios		Test subject		Accuracy		
	Indoor	Outdoor	Seen	Unseen	Precision	Recall	F1-Score
RF	●	○	●	○	75.2%	68.2%	72.1%
	●	○	○	●	46.7%	42.3%	44.5%
	○	●	●	○	71.3%	66.2%	69.5%
	○	●	○	●	40.3%	44.9%	43.7%
SVM	●	○	●	○	73.1%	71.5%	72.4%
	●	○	○	●	42.2%	45.1%	43.7%
	○	●	●	○	68.7%	72.3%	70.2%
	○	●	○	●	41.6%	46.2%	43.1%
CNN	●	○	●	○	72.5%	76.4%	74.4%
	●	○	○	●	52.1%	40.8%	45.9%
	○	●	●	○	77.3%	68.6%	73.8%
	○	●	○	●	43.7%	59.2%	50.6%
LSTM	●	○	●	○	78.2%	73.9%	76.1%
	●	○	○	●	57.8%	61.4%	59.5%
	○	●	●	○	75.1%	65.6%	70.3%
	○	●	○	●	52.7%	56.3%	54.4%
GPT4 - DO*	●	○	○	●	55.7%	62.1%	59.1%
	○	●	○	●	57.2%	59.3%	58.2%
GPT4 - CoT*	●	○	○	●	83.3%	80.1%	83.6%
	○	●	○	●	79.2%	74.5%	76.7%

format, we empower the Large Language Model (LLM) to freely generate a richer textual output. This approach leverages the model's inherent analytical capabilities to tap into and utilize the pertinent knowledge it has internalized.

B. Evaluation

We select GPT4 [20], the most advanced and powerful LLM currently accessible, to conduct a comprehensive analysis with four other baseline models across two datasets. Additionally, we compare the prompt mode of direct output (DO) without analysis to verify the effectiveness of applying CoT prompt for LLMs to improve prediction accuracy.

Overall Performance: The comprehensive test results are detailed in Tab. I. GPT4, by harnessing its robust comprehension capabilities along with the CoT prompt strategy, has delivered outstanding performance across all baseline assessments on the unseen dataset. It boasts an impressive average F1-score of 80.1%. A deeper analysis reveals that in comparison to its predecessor, GPT4-DO, the GPT4 model enhanced with CoT prompts—referred to as GPT4-CoT—shows a remarkable

ChatGPT

To interpret the given sequence, I'll compare the characteristics of the accelerometer, gyroscope, and magnetometer data with the examples provided for the turn around, turn left, and straight trajectories.

Given Sequence Interpretation:

1. **Accelerometer**: The x values show a significant change from -0.5 to 0.8 and then stabilize, while the y values start at -0.1, drop to -0.7, and then fluctuate around -0.2. The z values show an initial high of 0.4, drop to -0.3, and then go back to around 0.1. The accelerometer data suggests significant movements in all three axes, which is consistent with a turning or rotating motion rather than a straight path.

2. **Gyroscope**: The z-axis gyroscope data shows a peak at 1.0, followed by consistent values around 0.7 before dropping to 0.1. This indicates significant rotational movement around the z-axis at one point, which then stabilizes. This kind of pattern could suggest a turn left or a turn around, but the magnitude of change (peaking at 1.0) suggests the movement is more consistent with a turn around rather than just a turn left.

3. **Magnetometer**: The x and y magnetometer readings show large swings, which indicates a change in the orientation of the device relative to the Earth's magnetic field. The x value goes from 20.7 to -25.7, and the y value from -5.2 to 31.5, which suggests a substantial rotation, consistent with either a turn left or a turn around.

Comparing these readings with the examples:

- The **turn around** example shows significant changes in all three magnetometer axes, and large fluctuations in gyroscope z values.
- The **turn left** example shows moderate changes in the magnetometer values and smaller but noticeable changes in the gyroscope z values.
- The **straight** example shows relatively stable magnetometer values and minimal gyroscope z-axis rotation.

Given the data from the sequence, particularly the large changes in magnetometer readings and the significant motion indicated by the gyroscope z-axis, the movement pattern most closely resembles a **turn around** trajectory. The initial large change in the accelerometer's x-axis could be part of the maneuver to initiate the turn around. The subsequent stabilization suggests that after this rotation, the movement becomes more steady.

(a) GPT3.5

Claude

Certainly! Let's analyze the IMU data step by step to determine the robot's trajectory.

Step 1: Examine the accelerometer data
The accelerometer data shows relatively small values in all three axes (x, y, and z). This suggests that the robot is not experiencing significant linear acceleration or deceleration. Therefore, we can infer that the robot is moving at a relatively constant speed.

Step 2: Analyze the gyroscope data
The gyroscope data provides information about the robot's angular velocity. We can see that the x and y values are close to zero, indicating minimal rotation along those axes. However, the z-axis shows a notable change:
- From index 1 to 3, the z-axis values increase from -0.1 to 0.2, suggesting a gradual counterclockwise rotation.
- From index 3 to 5, the z-axis values increase significantly from 0.2 to 0.8, indicating a more rapid counterclockwise rotation.
- From index 5 to 7, the z-axis values decrease from 0.8 to 0.0, suggesting a slowing down of the counterclockwise rotation.

Step 3: Examine the magnetometer data
The magnetometer data measures the magnetic field strength and can provide information about the robot's orientation relative to the Earth's magnetic north. We can observe the following:
- The x-axis values start around 24 and gradually decrease to around -13, indicating a significant change in the robot's heading.
- The y-axis values also show a notable change, starting from -1.6 and decreasing to around -8.
- The z-axis values remain relatively consistent, suggesting no significant tilt or pitch of the robot.

Step 4: Deduce the robot's trajectory
Based on the analysis of the IMU data, we can conclude that the robot is performing a "turn left" trajectory. The key observations that support this conclusion are:
- The accelerometer data indicates relatively constant speed, suggesting no significant straight-line acceleration or deceleration.
- The gyroscope data, particularly the z-axis, shows a clear pattern of counterclockwise rotation, which aligns with a left turn.
- The magnetometer data confirms a significant change in the robot's heading, consistent with a turning motion.

The combination of minimal linear acceleration, significant counterclockwise rotation in the z-axis of the gyroscope, and the change in heading observed in the magnetometer data strongly suggests that the robot is executing a left turn.

In summary, based on the provided IMU data and the step-by-step analysis, the robot's trajectory can be classified as a "turn left" motion.

(b) Claude

GEMINI

Step 1: Examine the Accelerometer Data
The accelerometer readings provide information about the robot's linear acceleration. By analyzing the changes in acceleration, we can infer the robot's motion.

* The x-axis acceleration shows a pattern of alternating positive and negative values, indicating the robot is moving forward and backward.
* The y-axis acceleration shows a gradual decrease, suggesting the robot is turning left.
* The z-axis acceleration remains relatively constant, indicating no significant vertical movement.

Step 2: Analyze the Gyroscope Data
The gyroscope readings measure the robot's angular velocity. By integrating these readings, we can obtain the robot's orientation.

* The x-axis gyroscope shows no significant rotation.
* The y-axis gyroscope shows a gradual increase in negative values, confirming the left turn.
* The z-axis gyroscope shows no significant rotation.

Step 3: Consider the Magnetometer Data
The magnetometer readings provide information about the Earth's magnetic field. By comparing the magnetometer readings with the expected magnetic field at the robot's location, we can determine the robot's heading.

* The magnetometer readings show a gradual change in the x-axis, indicating the robot is turning left.
* The magnetometer readings in the y-axis and z-axis remain relatively constant.

Conclusion
Based on the analysis of the IMU data, we can conclude that the robot performed a **turn left** trajectory.

The key evidence for this conclusion is the gradual decrease in the y-axis acceleration and the corresponding increase in negative y-axis gyroscope readings. These readings indicate that the robot is rotating about its y-axis, which corresponds to a left turn.

The magnetometer readings also support this conclusion, as they show a gradual change in the x-axis, which is consistent with the robot turning left. Therefore, the robot's trajectory is **turn left**.
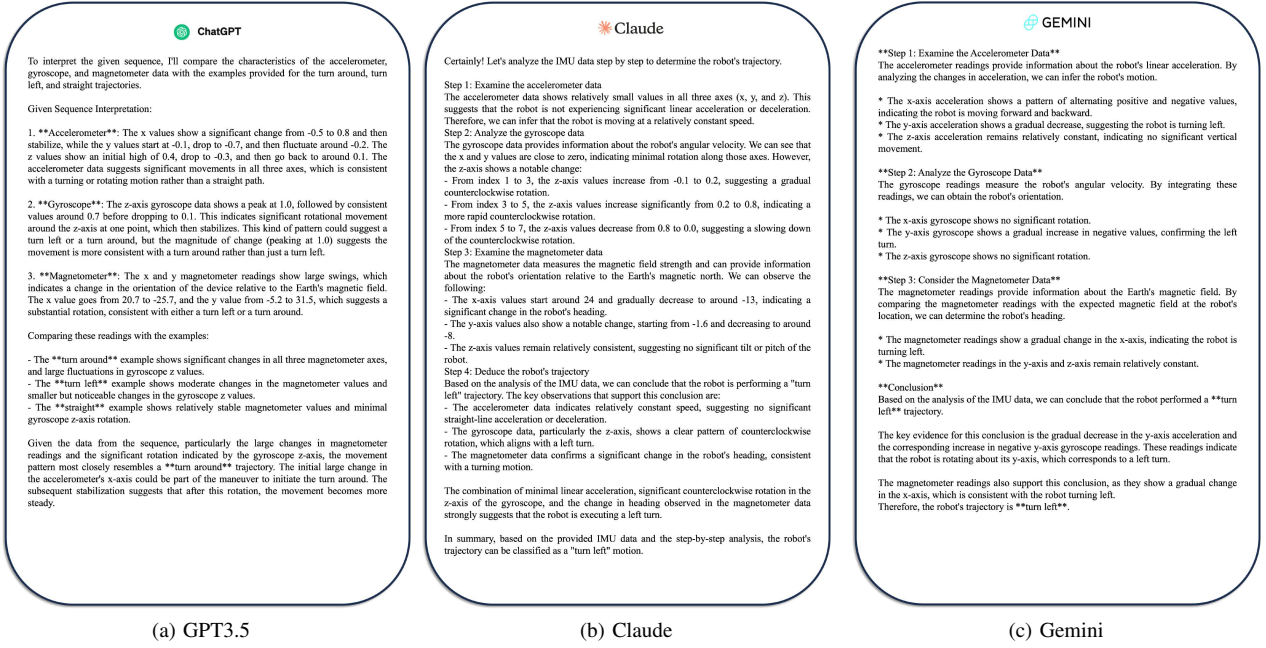
(c) Gemini

Figure 6: A comparison of the inference results generated by other LLMs for the turn-left scenario.

improvement of 38%. This significant enhancement underscores the value of contextualized prompting in achieving higher levels of model performance. In contrast, traditional neural network models like LSTM and CNN, while considered top-tier baselines, only manage to achieve a relatively modest average F1-score of approximately 73%. This suggests that while they remain useful, there is a substantial gap between their performance and the more advanced GPT4-CoT. The comparison becomes even more stark when evaluating the performance of two other conventional machine learning methods, RF and SVM. These methods show markedly poorer performance, hovering around the 70% mark. This performance disparity highlights the limitations of traditional machine learning techniques when faced with complex tasks that GPT4-CoT handles with relative ease.

Performance under Different Scenarios: We subsequently delved into an analysis of the model's performance across various environments. The data presented in Tab. I confirms that, for nearly all models evaluated, indoor performance surpasses outdoor performance. More precisely, with the GPT4-DO model, indoor performance exceeds outdoor by 2%. The discrepancy is even more pronounced with the GPT4-CoT model, where indoor results are 8% higher than those outdoors. This discrepancy can largely be attributed to the outdoor environment's greater variability in floor surfaces, as opposed to the more uniform and flat indoor surfaces, which result in fewer fluctuations in the IMU sensor data. In conclusion, the GPT4-CoT model demonstrates a robust capacity to achieve high performance in both indoor and outdoor scenarios, handling the varying conditions with commendable effectiveness.

Detailed Inference Example: To demonstrate the adeptness of GPT4 in generating expert-level insights and precise inferences, we provide a detailed representation of how GPT4 interprets the concept of a 'turn-left' maneuver in an indoor scenario, as illustrated in Fig. 5. The inference methodology is segmented into four distinct phases:

- Initially, GPT4 reviews and clarifies the provided information, ensuring a clear understanding of the problem at hand.
- In the subsequent phase, GPT4 employs embedded "expert knowledge" to accurately interpret the raw IMU sensor data that corresponds to specific movements. For instance, both 'turn-left' and 'turn-right' maneuvers are associated with variations in the gyroscope readings, 'go-straight' exhibits minimal fluctuation, and 'turn-around' is characterized by more pronounced changes in gyroscope values.
- The third phase involves GPT4 conducting a more granular analysis of the input raw data, which leads to the identification that the data is indicative of a particular pattern.
- Finally, after a thorough analysis and integration of the previous expert insights, GPT4 arrives at a well-founded conclusion in the fourth stage, determining that the action is most likely a 'turn-left' trajectory.

This step-by-step process exemplifies GPT4's ability to not only process raw sensor data but also to apply specialized knowledge for accurate trajectory tracing.

III. DISCUSSION

Logical Reasoning Ability: Previous studies have confirmed

that LLMs are equipped with the potential to perform logical deductions. Moreover, our studies have revealed that their proficiency is not limited to textual data interpretation but also to making sense of data from sensory inputs. Models like GPT4 have shown remarkable adaptability in translating raw sensory data into conceptual verbal constructs, identifying patterns as intermittent, static, or sudden. This capability is analogous to a powerful sieve that effectively tackles the issue of out-of-distribution data, a notable challenge for standard machine learning or deep learning models. LLMs prioritize the creation of credible narratives that mirror human cognitive patterns, shifting away from a sole reliance on mimicking the exact features of raw inputs. Nonetheless, there are variances in the logical deductive strengths across different LLMs. The comparisons in Fig. 6 illustrate the variations in response to the same prompt when tested on other models such as GPT3.5, Anthropic Claude, and Google Gemini. It is important to highlight that the presence of reasoning skills in a model does not directly imply higher intelligence. Notably, some smaller-scale LLMs like GPT3.5 and Google Gemini have shown to lag behind GPT4 in their reasoning accuracy, despite being prompted in the same way.

IV. FUTURE WORK AND CONCLUSIONS

The results of this research indicate that LLMs can serve as a primary framework for accurately and reliably mapping movement paths in a zero-shot context. This suggests that LLMs have the innate ability to interpret data from IoT sensors without relying on example-based guidance from experts. The study underscores the potential of LLMs to process unrefined sensor data, signaling a significant potential shift in the AIoT field. However, it is essential to conduct further research to determine the specific conditions and limits of LLM effectiveness. The establishment of a more rigorous and comprehensive set of evaluation methods and standards is necessary. Enhancing our understanding of the strengths and limitations of Language Models will enable us to utilize their full potential in analyzing complex real-world data, such as abstract Channel State Information in wireless devices.

REFERENCES

[1] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun *et al.*, "Personal llm agents: Insights and survey about the capability, efficiency and security," *arXiv preprint arXiv:2401.05459*, 2024.

[2] X. Zhang, R. R. Chowdhury, R. K. Gupta, and J. Shang, "Large language models for time series: A survey," 2024.

[3] Y. Li, Z. Li, P. Wang, J. Li, X. Sun, H. Cheng, and J. X. Yu, "A survey of graph meets large language model: Progress and future directions," 2024.

[4] T. Brooks, B. Peebles, C. Homes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh, "Video generation models as world simulators," 2024. [Online]. Available: https://openai.com/research/video-generation-models-as-world-simulators

[5] Claude. (2024, March). [Online]. Available: https://www.anthropic.com/news/claude-3-family

[6] Y. LeCun, "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27," *Open Review*, vol. 62, 2022.

[7] S. Ji, X. Zheng, and C. Wu, "Hargpt: Are llms zero-shot human activity recognizers?" 2024.

[8] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, 2016.

[9] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[10] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition." in *Ijcai*, vol. 15. Buenos Aires, Argentina, 2015, pp. 3995–4001.

[11] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, "Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 220–233.

[12] K. Irie, Z. Tüske, T. Alkhouli, R. Schlüter, H. Ney *et al.*, "Lstm, gru, highway and a bit of attention: An empirical overview for language modeling in speech recognition," in *Interspeech*, 2016, pp. 3519–3523.

[13] H. Yang, H. Liu, C. Luo, Y. Wu, W. Li, A. Y. Zomaya, L. Song, and W. Xu, "Vehicle-key: A secret key establishment scheme for lora-enabled iov communications," in *IEEE ICDCS*, 2022.

[14] W. Xu, H. Yang, J. Chen, C. Luo, J. Zhang, Y. Zhao, and W. J. Li, "Washring: An energy-efficient and highly accurate handwashing monitoring system via smart ring," *IEEE TMC*, 2024.

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[16] H. Nori, Y. T. Lee, S. Zhang, D. Carignan, R. Edgar, N. Fusi, N. King, J. Larson, Y. Li, W. Liu *et al.*, "Can generalist foundation models outcompete special-purpose tuning? case study in medicine," *arXiv preprint arXiv:2311.16452*, 2023.

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[18] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[19] Y. Kim, X. Xu, D. McDuff, C. Breazeal, and H. W. Park, "Health-llm: Large language models for health prediction via wearable sensor data," *arXiv preprint arXiv:2401.06866*, 2024.

[20] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.