By My Eyes: Grounding Multimodal Large Language Models with Sensor Data via Visual Prompting

Hyungjun Yoon[1], **Biniyam Aschalew Tolera**[1],
Taesik Gong[2], **Kimin Lee**[1], **Sung-Ju Lee**[1]

[1]KAIST [2]UNIST

{hyungjun.yoon, binasc, kiminlee, profsj}@kaist.ac.kr, taesik.gong@unist.ac.kr

Abstract

Large language models (LLMs) have demonstrated exceptional abilities across various domains. However, utilizing LLMs for ubiquitous sensing applications remains challenging as existing text-prompt methods show significant performance degradation when handling long sensor data sequences. We propose a visual prompting approach for sensor data using multimodal LLMs (MLLMs). We design a visual prompt that directs MLLMs to utilize visualized sensor data alongside the target sensory task descriptions. Additionally, we introduce a visualization generator that automates the creation of optimal visualizations tailored to a given sensory task, eliminating the need for prior task-specific knowledge. We evaluated our approach on nine sensory tasks involving four sensing modalities, achieving an average of 10% higher accuracy than text-based prompts and reducing token costs by 15.8×. Our findings highlight the effectiveness and cost-efficiency of visual prompts with MLLMs for various sensory tasks. The source code is available at https://github.com/diamond264/ByMyEyes.

1 Introduction

Large language models (LLMs) have shown remarkable performance in tasks across diverse domains, including science, mathematics, medicine, and psychology (Bubeck et al., 2023). The recent advent of multimodal LLMs (MLLMs), *e.g.*, GPT-4o (OpenAI, 2024), has further expanded their capabilities to images and audio inputs, broadening their use in fields such as industry and medical imaging (Yang et al., 2023).

Meanwhile, sensor data—including measurements from smartphones, wearables, IoT (Dian et al., 2020), and medical equipment (Pantelopoulos and Bourbakis, 2009)—holds potential for ubiquitous applications when effectively integrated with MLLMs. Sensory tasks involve extensive and

Figure 1: An example of solving a sensory task using an MLLM with visual prompts. The visualization generator generates an appropriate visualization for the given sensor data, and the visualized data is provided as an image to the MLLM for solving the task.

significant applications, ranging from authentication (Abuhamad et al., 2020) and healthcare (Wang et al., 2019) to agriculture (Sishodia et al., 2020) and environmental monitoring (Feng et al., 2019). However, MLLMs remain underutilized. The diversity of sensors (Wang et al., 2019) and the heterogeneity among them (Stisen et al., 2015) hinder the implementation of a foundational model that generalizes across various sensing tasks. The expensive data collection (Vijayan et al., 2021) often results in insufficient training data, further complicating the development of such capability.

Recent studies explored leveraging pre-trained LLMs to solve general sensory tasks (Yu et al., 2023; Liu et al., 2023; Kim et al., 2024). One approach extracts task-specific features from sensor data and composes them as prompts (Yu et al.,

2023). However, designing such prompts requires specific domain knowledge. Alternatively, incorporating raw sensor data as text prompts (Kim et al., 2024; Liu et al., 2023) has been a widely used method to handle sensory data with LLMs as a more generalizable solution. Yet, we empirically found that providing raw sensor data with text prompts shows poor performance in real-world sensory tasks with long-sequence inputs and incurs high costs due to an extensive number of tokens.

To address these challenges, we propose providing visualized sensor data as images to MLLMs that support visual inputs. Leveraging MLLMs' growing ability to interpret visual aids (Yang et al., 2023), we explore their effectiveness in analyzing plots generated from sensor data. We designed a visual prompt comprising visualized sensor data and task-specific instructions to solve sensory tasks. In addition, we present a visualization generator that enables MLLMs to independently generate optimal visualizations using tools available in public libraries. This generator filters potential visualization methods based on the task description and assesses the resulting visualizations of each method to determine the best visualization. Figure 1 compares the existing text-only prompts with our method for sensory tasks.

Evaluations on nine sensory tasks involving four different modalities showed that visual prompts generated from the visualization generator significantly improved performance by an average of 10% while reducing token costs by $15.8\times$ compared with the existing baseline. Our findings highlight the effectiveness and efficiency of visualized sensor data with MLLMs in various applications.

We summarize our contributions as follows:

- We propose to ground MLLMs with sensor data by providing visualized sensor data as images, achieving improved performance at reduced costs than the text-based baseline.

- We present a visualization generator that automatically generates suitable visualizations for various sensory tasks using public libraries.

- We conduct experiments on nine different sensory tasks across four modalities, demonstrating the broad applicability of our approach.

2   Related Work

LLMs with sensor data. Sensory tasks involve sequences of numbers indicating values over time.

Initial research for handling sequential data focused on time-series forecasting (Zhang et al., 2024c). Converting time-series data into text prompts for forecasting has been proposed in PromptCast (Xue and Salim, 2023) and LLMTime (Gruver et al., 2024). Other studies (Zhou et al., 2023; Jin et al., 2023a) used specialized encoders to create embeddings compatible with pre-trained LLMs.

Beyond forecasting, LLMs have been explored in healthcare for their ability to answer questions using physiological sensor data (Liu et al., 2023). For example, LLMs have been used for ECG diagnosis (Yu et al., 2023) by integrating ECG-specific features and knowledge from ECG databases. Penetrative AI (Xu et al., 2024) and Health-LLM (Kim et al., 2024) have used raw sensor data in text prompts to solve health problems without task-specific processing. Our study examines whether existing methods can generalize to broader sensing tasks with high-frequency, long-duration data. Building upon these works, we propose visualizing sensor data for MLLMs to improve their performance and cost efficiency.

Multimodal large language models (MLLMs). Advancements in MLLMs (Zhang et al., 2024a) have equipped popular models such as ChatGPT (OpenAI, 2022) with vision capabilities (OpenAI, 2024). Recent studies explored the in-context learning (Brown et al., 2020) abilities of MLLMs, showing that they can understand images with the interleaved text and few-shot examples (Tsimpoukelli et al., 2021; Alayrac et al., 2022). This capability has been applied in medical diagnostics, including analyzing radiology and brain images with accompanying text instructions (Wu et al., 2023). Our work explores using MLLMs to analyze visualized sensor data for broader applications.

Using tools with LLMs. Recent research has shown that augmenting LLMs with external tools can extend their capabilities. Toolformer (Schick et al., 2024) enables LLMs to access public APIs and search engines, while Visual Programming (Gupta and Kembhavi, 2023) uses LLMs to generate and execute codes. HuggingGPT (Shen et al., 2024) and Chameleon (Lu et al., 2024) integrated multiple expert models to enhance functionalities. Recently, Data Interpreter (Hong et al., 2024) enabled LLMs to analyze data and build task-specific models for data interpretation. Building on them, our work leverages MLLMs to utilize sensor data visualization tools. Importantly, unlike existing approaches that rely on external tools as the pri-

mary task solvers, we propose positioning MLLMs as the main solvers for sensory tasks, leveraging their in-context learning abilities. Our approach aims to eliminate the need for additional data collection and training, thereby addressing the scarcity of public resources for sensory tasks. Furthermore, we introduce a design in which MLLMs perform demonstration-based assessments to evaluate their task-solving effectiveness, ensuring optimal visualization for specific tasks.

3 Limitations of Representing Sensor Data as Text-based Prompts

Existing approaches for grounding language models with sensor data primarily rely on text-based prompts (Liu et al., 2023; Jin et al., 2023b; Zhang et al., 2024c; Yu et al., 2023). One approach uses prompts with specialized features extracted from sensor data for specific tasks, such as R-R intervals for ECG-based applications (Yu et al., 2023). While this approach effectively handles known sensory tasks, feature-based prompts often require domain knowledge, which is not generalizable for non-expert users. Instead, a more common approach (Kim et al., 2024; Xu et al., 2024; Liu et al., 2023) incorporates raw sensor data sequences directly into prompts without data processing. However, most studies focus on short sequences (e.g., fewer than 100 elements) (Kim et al., 2024) and simple tasks (e.g., binary classification) (Liu et al., 2023).

Real-world sensor data often entail long numeric sequences with high sampling rates and long durations. For example, arrhythmia detection (Wagner et al., 2020) requires ECG data sampled at 100Hz over 10 seconds, resulting in 1,000 elements. This section investigates the limitations of using text-based prompts to represent such complicated sensor data in language models. We focus on the capability to interpret sensor data and the token consumption costs associated with long numeric sequences.

Language models struggle to interpret long numeric text sequences. Language models interpret simple numeric sequences by performing arithmetic operations (Achiam et al., 2023) and understanding sequential data (Gruver et al., 2024; Mirchandani et al., 2023). However, we empirically revealed that their performance declines significantly with longer sequences inside the prompt, such as those exceeding 100 numbers, common in
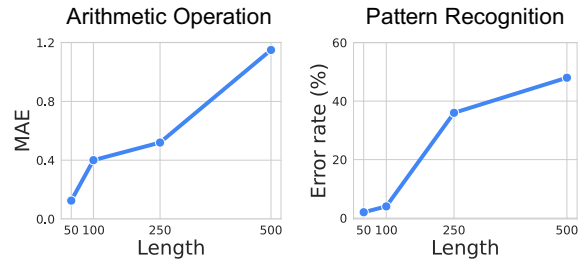

Figure 2: Performance of GPT-4o on arithmetic operation (mean prediction) and pattern recognition (sine and sawtooth wave classification) tasks for varying lengths.

sensor data.

We conducted experiments with two specific tasks: *mean prediction* to evaluate arithmetic capabilities (Pirttikangas et al., 2006) and *wave classification* to assess pattern recognition (Liu et al., 2016) in sequences. The defined tasks represented the basic functionalities for sensor data interpretation, serving as typical feature extraction methods. Using randomly generated sine and sawtooth waves with varying lengths, we asked a language model, GPT-4o (OpenAI, 2024), to calculate mean values and classify wave types using one-shot examples for each task. Each task was repeated 30 times to ensure robustness.

Figure 2 shows the results. In arithmetic operations, error rates consistently increased with the length. In pattern recognition, performance declines significantly for sequences longer than 100 elements, approaching the performance of a random classifier at 500 elements. While recent models such as GPT-4o, with its 128K context window, support long input lengths, our results indicate that interpreting sensor data with long numeric sequences still remains challenging.

Sensor data in text is costly. The computational and financial burdens for API users of language models scale with the number of tokens in the prompt. Representing sensor data in textual format leads to extensive token usage, thereby increasing costs. For instance, performing passive sensing to track activities with smartphone accelerometers (Stisen et al., 2015) uses sensor data sampled at 100 Hz. Collecting this data over a minute results in a prompt of 18K numbers, translating to 90K tokens. This leads to a huge cost of $450 per hour when using GPT-4o API to classify six activities with one example for each. Higher sampling rates or longer durations further increase the costs, making such applications infeasible.
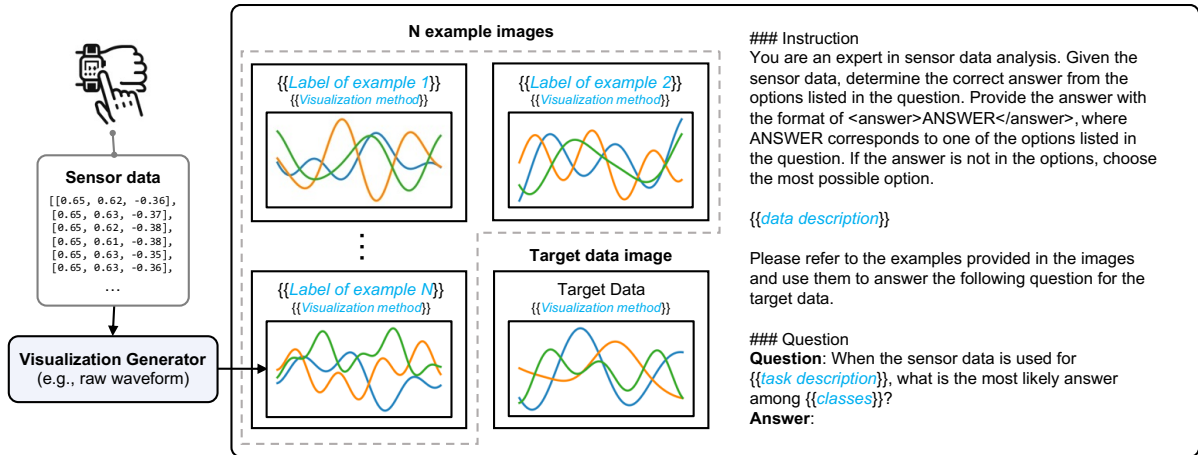
Figure 3: Overview of our visual prompt. Sensor data are transformed into annotated images with labels and visualization methods. Additionally, instructions are provided to the MLLM, detailing the task and relevant data descriptions. These instructions guide the MLLM on effectively utilizing the provided images to solve the task.

Transition to visual prompts. Language models such as ChatGPT (GPT-4o (OpenAI, 2024)) and Gemini (DeepMind, 2024) have expanded capabilities to include multimodal inputs (e.g., vision and audio). Recent Multimodal Large Language Models (MLLMs) demonstrate an increasing ability to identify patterns and interpret visual data (Achiam et al., 2023). This opens new opportunities for sensory tasks, as sensor data are often visualized for analysis. Visualizations make complex data more interpretable and condense long data sequences into a *single* image, significantly reducing token costs. Building on this capability, we exploit visualized sensor data instead of text-based prompts.

4 Method

We introduce our method for handling sensory tasks by providing sensor data as image inputs to MLLMs. Section 4.1 overviews our prompt design strategy. Section 4.2 introduces our visualization generator, which automatically generates suitable visualizations for heterogeneous sensor data.

4.1 Visual Prompt Design

To leverage MLLMs for sensory tasks, we propose a *visual prompt*, as illustrated in Figure 3. The key idea is to transform numeric sequences of sensor data into visual plots using various methods, such as raw waveforms and spectrograms. Detailed information about these visualization methods is in Section 4.2. For few-shot examples, each plot includes a label as a title above it (i.e., *{{Label of example X}}*). For unlabeled target data used in queries, the title is simply stated as "target data".

We provide textual instructions to clarify the data collection process and the task's objectives. These instructions ensure that MLLMs can effectively interpret and utilize the visualized sensor data.

4.2 Visualization Generator

In our proposed visual prompt, the choice of visualization method is crucial, as it significantly influences the MLLM's ability to comprehend the sensor data. For example, raw waveform plots are ideal for tasks involving amplitude pattern recognition over time, while spectrograms (Ito et al., 2018) are suitable for tasks relying on frequency features. We introduce a *visualization generator* that automatically chooses the most suitable visualization tool from available public libraries, enabling non-expert users to effectively utilize visual prompts. This generator operates in two main phases: (i) visualization tool filtering and (ii) visualization selection (see Figure 4).

Visualization tool filtering. Public libraries offer a vast array of sensor data visualizations. However, trying each out to identify the optimal visualization is computationally expensive. To minimize the cost, we employ a filtering approach. By providing available visualization tools, descriptions of the task, and data collection, we ask MLLMs to select a list of visualization methods useful for the target task.

As shown in Figure 4 (green box), we provide a full list of available visualization methods found in public libraries (*e.g.*, Matplotlib (Hunter, 2007), Scipy (Virtanen et al., 2020), and Neurokit2 (Makowski et al., 2021)) along with task and
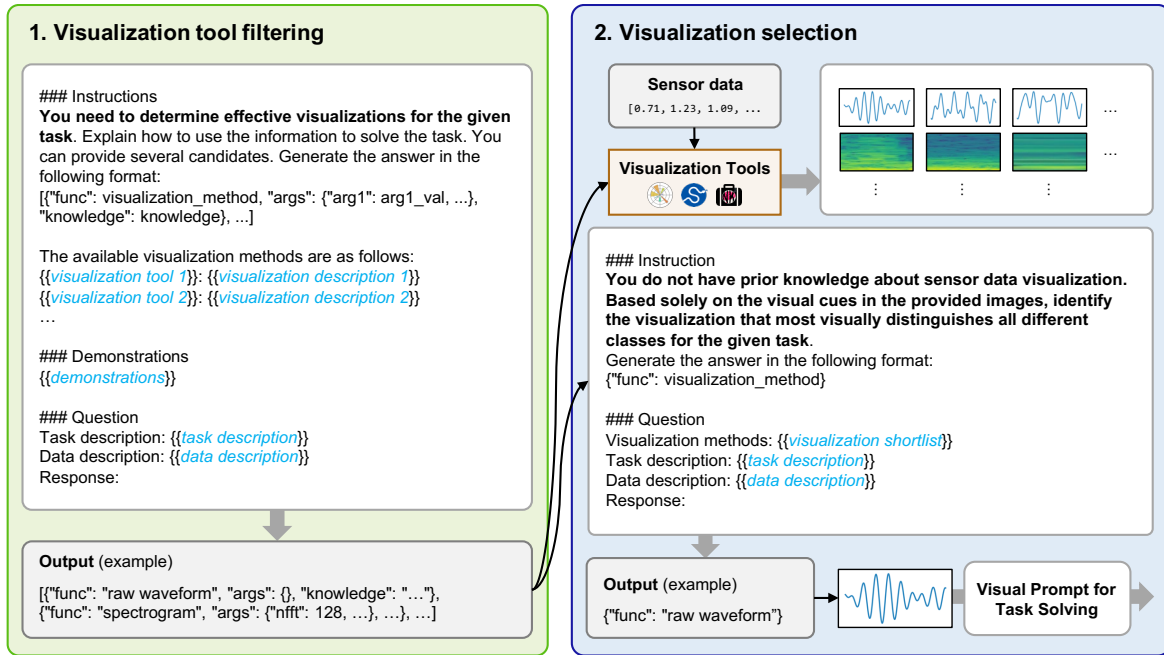
Figure 4: Overview of our visualization generator. First, visualization tool filtering generates a filtered list of visualization tools from public libraries based on the task and data descriptions. Next, visualization selection generates and selects the most effective visualization by asking MLLMs to observe visualized sensor data prepared for the task using all the filtered visualization methods.

data collection descriptions to MLLMs. We also leverage the in-context learning ability of MLLMs to enhance response quality by providing demonstrations of optimal visualizations chosen for different tasks. The MLLM is instructed to output a list in JSON format, which is suitable for automated parsing at a later stage. Appendix E shows the full list of available visualization tools and demonstrations.

Visualization selection Sensor data exhibit variations by instance due to user-specific behaviors, environmental factors, or device settings (Stisen et al., 2015), which cannot be fully captured in task and data descriptions. This variability limits the reliability of selecting visualizations based solely on the descriptions. To address this, we visualize the sensor data using all filtered visualization tools and ask the MLLM to select the one that provides the best visual information for the task.

The blue box in Figure 4 illustrates this procedure. First, different visualizations are generated using the filtered tools. With the images, we instruct the MLLM to select the best visualization by providing a textual prompt, including the visualization methods, task, and data details. We found that MLLM often makes incorrect decisions by prioritizing the task description over the visual aids. To prioritize visual efficacy, we explicitly instruct

the MLLM to avoid relying on prior knowledge about sensor data and focus on the provided images. Finally, our automated framework conveys the selected visualization to the visual prompt for task solving.

5 Experiments

We evaluate the applicability of our approach with MLLMs by conducting experiments on a range of sensory tasks.

5.1 Setups

We assume a practical scenario where non-expert users attempt to solve sensory tasks using MLLMs (1) without prior knowledge of relevant features and (2) without external resources to fine-tune the MLLM. Given the constraints, we leveraged the few-shot prompting (Brown et al., 2020) approach. For the main evaluation, we used 1-shot examples where users provide the MLLM with minimal examples to guide task-solving.

Sensory tasks We established nine different sensory tasks across four sensor modalities: accelerometer, electrocardiography (ECG) sensor, electromyography (EMG) sensor, and respiration sensor. We used three datasets for tasks using accelerometers: HHAR (Stisen et al., 2015) for basic human activity recognition (running and walking),

Table 1: Comparison of the text prompts and visual prompts for solving sensory tasks using GPT-4o. The highest accuracy values are highlighted in bold. The visual prompts (multi-shot) utilize the maximum number of examples by matching the token size of the 1-shot text prompts.

| | Accelerometer | | | ECG | | | | EMG | Resp | |
Method	HHAR	UTD-MHAD	Swim	PTB-XL (CD)	PTB-XL (MI)	PTB-XL (HYP)	PTB-XL (STTC)	Gesture	WESAD	Avg.
Accuracy										
Task-specific model	0.95	0.95	0.99	0.88	0.86	0.90	0.90	0.64	0.69	0.86
Text-only prompt	0.66	0.10	0.51	0.73	0.62	0.47	0.53	0.27	0.48	0.49
Visual prompt (ours)	**0.67**	**0.43**	**0.73**	**0.80**	**0.68**	**0.55**	**0.57**	**0.30**	**0.61**	**0.59**
Number of tokens										
Text-only prompt	52910	50439	16586	3204	2766	2757	3596	88655	60253	31244
Visual prompt (ours)	2020	5963	1768	943	943	943	946	3073	1211	1979
	(26.2×↓)	(8.5×↓)	(9.4×↓)	(3.4×↓)	(2.9×↓)	(2.9×↓)	(3.8×↓)	(28.9×↓)	(49.8×↓)	(15.8×↓)

UTD-MHAD (Chen et al., 2015) for complex activity recognition with fine-grained arm motions, and a swimming style recognition dataset (Brunner et al., 2019). We use the PTB-XL (Wagner et al., 2020) dataset for the arrhythmia diagnosis tasks that use ECG. The dataset includes detection tasks for four different types of arrhythmia symptoms. For EMG data, we used a dataset (Ozdemir et al., 2022) for hand gesture recognition. Finally, we used a stress detection task using respiration sensors provided by the WESAD (Schmidt et al., 2018) dataset. Details on each task, including the classes, sampling rates, windowing durations, and specific configurations, are in Appendix F.

Data processing. We normalized data using the mean and standard deviation values calculated for each user. Test splits were created by randomly sampling 30 samples per class. For the UTD-MHAD dataset, we sampled 10 samples per class due to the limited sample availability. Examples of few-shot prompting were randomly sampled, ensuring no overlap with the test set. Each task employed the window sizes and sampling rates specified in the original dataset descriptions (see Appendix F).

Baselines. We set *text-only prompts* for conveying sensor data to MLLMs as the main baseline to be compared with our visual prompts. Text-only prompts represented sensor data as numbers within the prompt. We designed text-only prompts by following the latest prompting studies incorporating sensor data into LLMs for healthcare (Liu et al., 2023). Additionally, to establish an upper bound for task-specific performance, we included a fully-supervised baseline using neural networks trained on 75% of the entire data after excluding the test and validation sets. We adopted architectures widely accepted for each type of sensor data:

1D CNNs for activity recognition with accelerometers (Chen et al., 2021) and EMG data (Xiong et al., 2021), as well as for WESAD (Vos et al., 2023), and XResNet-101 for PTB-XL (Strodthoff et al., 2023).

Implementation. We used GPT-4o from the OpenAI API (OpenAI, 2024) as MLLM. The text-only prompts contained the same information as the generated visualization to ensure a fair comparison between text-only and visual prompts. For example, if the visualization generator outputs a plot with peak notations, the corresponding text-only prompt contains the same features, including the peak values with their indices. When the information could not fit within the token limit (128K), we used the raw waveform.

Metrics. We evaluated the experimental results based on accuracy. We also assess the number of tokens used by each prompt method. Tokens are counted using the o200k_base encoding used for GPT-4. To estimate the token cost for images in the same space as text, we follow the computation guidelines provided by OpenAI (OpenAI, 2024).

5.2 Results

Performance. Table 1 shows the overall performance of utilizing visual prompts for solving sensory tasks. For the same 1-shot prompting, visual prompts consistently showed enhanced accuracies than text-only prompts, achieving an average increase of 10%. Notably, the UTD-MHAD task exhibited a significant accuracy gain of up to 33%. See Appendix G for prompt examples with resulting visualizations.

In addition to achieving higher accuracy, visual prompts are more cost-effective. The number of tokens used for visual prompts in Table 1 shows a
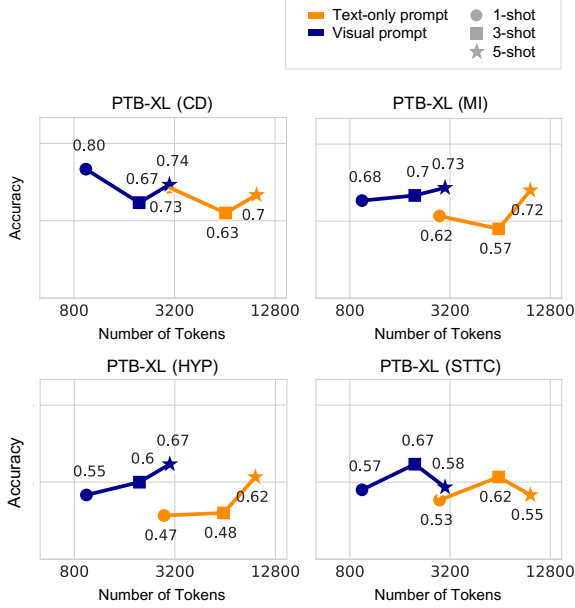
Figure 5: Accuracy of arrhythmia detection tasks using visual and text-only prompts with different shots.

substantial reduction, averaging $15.8\times$ fewer than text-only prompts. MLLMs calculate token costs for images within the same token space as text but with distinct counting criteria. In our experiments, GPT-4o counts tokens for images based on the number of 512×512 pixel blocks (N) covering the image input, calculated at $85 + 170 \times N$. Our visualized sensor data was represented within a single 512×512 pixel image, regardless of the sensor data length, significantly reducing costs. Note that the number of tokens from visual prompts is only affected by the number of examples, as all images are the same size. In contrast, text prompts are heavily influenced by high sampling rates and long durations.

To further understand the effectiveness of visual prompts with small tokens, we analyzed the information capacity at the same token cost. Considering a budget of 500 tokens, text-based prompts can include approximately 2,000 ASCII characters. In contrast, visual prompts can input two 512×512 px images. In terms of bytes, 2,000 ASCII characters amount to 2 KB, whereas two RGB images occupy 1.57 MB, which is $785\times$ larger. Although this calculation does not directly translate to the exact amount of useful information, it suggests that well-designed visual prompts can convey a wider range of information than text prompts within the same cost constraint.

Effect of number of examples. To investigate the impact of varying numbers of examples, we ex-

perimented using different numbers of examples (1-shot, 3-shot, and 5-shot) within the prompts. We used the ECG dataset, allowing multiple examples with text-only prompts due to its lower token consumption.

Figure 5 depicts the results. Prompting methods are color-coded (blue for visual and green for text-only), and different markers indicate the number of shots. We compared the accuracy and counted the tokens for each setting. We found that visual prompts constantly outperformed text-only prompts with the same number of examples, indicating the robustness of our method in different few-shot examples.

Additionally, when comparing visual prompts and text-only prompts under the same token budget (5-shot visual prompt versus 1-shot text-only prompt), visual prompts often performed significantly better (MI and HYP detection). This highlights the advantage of token-efficient visual prompts that can utilize more resources for better performance under the same token constraint.

Unlike our expectations, additional examples did not always result in better performance. This result aligns with existing reports indicating that more examples do not always guarantee better results (Perez et al., 2021; Lu et al., 2021). We further hypothesize that a longer context might hinder the MLLM's ability to retrieve important information (Liu et al., 2024b). Our findings suggest that the impact of shots is data-dependent, and effectively utilizing more examples for consistent improvement remains an open question for further research. Note that our visual prompts consistently outperformed text prompts, even on datasets where additional shots negatively affect performance. This supports that the main improvement of our approach stems from using the visual modality for data interpretation, not merely from the token length reduction.

Effect of visualization generator. We conducted an ablation study to assess the impact of the visualization generator. We compared the visualization generator against two different baselines: (1) using a fixed visualization that defaults to raw waveform plots, and (2) a method that selects visualizations based solely on a text description of the task and data. For the second baseline, we utilized our visualization tool filtering prompt (see Appendix G for an example) to generate a single visualization, rather than filtering multiple tools.

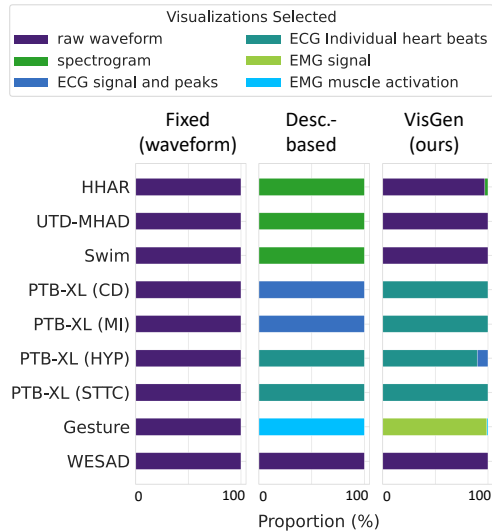Figure 6 summarizes the visualizations selected

Figure 6: Proportion of selected visualization methods from the baselines and our visualization generator across different tasks.
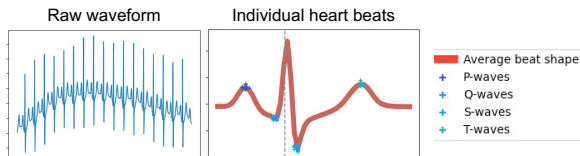

Figure 7: Examples of ECG visualizations. The visualization generator selected the individual heartbeats plot.

Table 2: Performance of using different visualization methods for visual prompts. We compare a fixed raw waveform plot (Fixed), visualizations selected based solely on a text description (Desc.-based), and visualizations from our visualization generator (VisGen). Ours is highlighted in blue cells, and performances from visualizations on certain tasks that show significant performance drops more than 10% compared to the highest are colored red.

	Visualization Method		
Dataset	Fixed (waveform)	Desc.-based	VisGen (ours)
Accelerometer			
HHAR	0.70	0.34	0.67
UTD-MHAD	0.41	0.05	0.43
Swim	0.74	0.20	0.73
ECG			
PTB-XL (CD)	0.60	0.69	0.80
PTB-XL (MI)	0.58	0.65	0.68
PTB-XL (HYP)	0.53	0.52	0.55
PTB-XL (STTC)	0.53	0.57	0.57
EMG			
Gesture	0.30	0.31	0.30
Respiration			
WESAD	0.62	0.60	0.61

by the baselines and our visualization generator. The selection from the description-based method (Desc.-based) and our visualization generator (VisGen) varied primarily based on sensor modalities. Our visualization generator mainly selected raw waveforms, occasionally spectrograms for accelerometer tasks (the same for WESAD). For ECG and EMG datasets, it selected specialized visualizations, such as ECG individual heart beats plot. The description-based method also selected modality-aware visualizations, but these differed from those chosen by ours. For instance, it selected spectrograms for the accelerometer tasks and signal and peak plots for ECG datasets. The key difference stemmed from whether the MLLM referenced the visualized image itself. Note that within each dataset, the target task and data collection protocol was consistently controlled, so the visualizations selected for samples within each dataset remained almost identical, despite our design allowing for sample-wise visualization selection.

The performance comparison of different visualization selection methods is shown in Table 2. Overall, our visualization generator achieves the best or comparable performance. In contrast, the baseline methods show significant performance degradation in certain tasks. For example, using a fixed raw waveform leads to a significant 20% performance drop for ECG tasks, as raw waveforms fail to provide insightful visual insights due to the complex structure of ECG (Figure 7). This illustrates that a fixed visualization cannot be generalized across different sensory tasks.

Similarly, the description-based method faces challenges with accelerometer tasks. It selects spectrograms, likely due to the world knowledge from public datasets, which may lead the MLLM to consider frequency features as the optimal information for motion data analysis. However, the dense and complex features in spectrogram images were difficult for the MLLM to interpret, leading to near-random performances. In contrast, our visualization generator compares visualized images, consistently avoiding suboptimal choices such as spectrograms for accelerometer tasks. This self-assessment mechanism ensures that the visualization generator selects the optimal visualization method among the possible options.

6 Conclusion

We addressed sensory tasks by providing visualized sensor data as images to MLLMs. We designed a visual prompt to instruct MLLMs in using visualized sensor data, provided with textual descriptions of the task and data collection methods. Additionally, we introduced a visualization generator that automatically selects the best visualization method for each task using visualization tools available in public libraries. We conducted experiments across nine different sensory tasks and four sensor modalities, each with a distinct task. Our results suggest that the visual prompts generated by our visualization generator not only improve accuracy by an average of 10% over text-based prompts but also significantly reduce costs, requiring $15.8\times$ fewer tokens. This indicates that our approach with visual prompts and a visualization generator is a practical solution for general sensory tasks.

Acknowledgements

Limitations

Our study demonstrates the effectiveness of visual prompts on nine different sensory tasks, primarily focusing on classification. While visual prompts effectively highlight patterns over images, for tasks requiring numerical retrieval or precise computations—where exact values are critical—text prompts can be more effective due to their inclusion of specific numeric data, which are omitted in visual representations. Notably, our approach integrates both images and texts in prompts, allowing the inclusion of numerical values in the text. Determining the optimal distribution of information between images and text to compose a prompt that effectively addresses sensory tasks presents a future direction for this work. Moreover, the inclusion of numeric values can result in long prompts, which affect both cost and performance. Extracting only the useful information, such as statistics or specific time splits, requires further research.

Visualizing sensor data as plots often presents challenges. For instance, brain wave analysis using high-density EEG involves up to 256 channels (Fiedler et al., 2022), complicating their representation in a single visual plot. We denote different channels as distinct notations within a plot, making densely populated plots visually indecipherable. An alternative method of plotting distinct channels across separate subplots was explored but resulted in a significant drop in performance (see Appendix 6). We hypothesize that this limitation arises from the dispersion of information across various areas, highlighting that effective visualization of large-channel datasets remains challenging. This underscores the need for improved visualization techniques in such scenarios.

Our visual prompt design does not incorporate Chain-of-Thought (CoT) prompting (Kojima et al., 2022). Experiments using zero-shot CoT on our datasets revealed inconsistent benefits (see Appendix A), unlike the widely known effect of CoT for enhancing performance. We suspect this may be due to the complexities of reasoning over sensory data. Given the observation, further research is needed to develop methods that effectively integrate reasoning and interpretation into the decision-making processes for sensor data analysis.

Lastly, the high costs of text-only prompts in sensory tasks constrained our testing to 30 samples per class. Expanding the scale as resources allow could provide a more robust analysis and potentially validate a broader spectrum of applications.

References

Mohammed Abuhamad, Ahmed Abusnaina, DaeHun Nyang, and David Mohaisen. 2020. Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey. *IEEE Internet of Things Journal*, 8(1):65–84.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language

model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Kerem Altun and Billur Barshan. 2010. Human activity recognition using inertial/magnetic sensor units. In *Human Behavior Understanding: First International Workshop. Proceedings 1*, pages 38–51. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Gino Brunner, Darya Melnyk, Birkir Sigfússon, and Roger Wattenhofer. 2019. Swimming style recognition and lap counting using a smartwatch and deep learning. In *ACM International Symposium on Wearable Computers*, pages 23–31.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. 2015. Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *International conference on image processing (ICIP)*, pages 168–172. IEEE.

Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40.

DeepMind. 2024. Gemini. https://deepmind.google/technologies/gemini/.

F John Dian, Reza Vahidnia, and Alireza Rahmati. 2020. Wearables and the internet of things (iot), applications, opportunities, and challenges: A survey. *IEEE access*, 8:69200–69211.

Shaobin Feng, Fadi Farha, Qingjuan Li, Yueliang Wan, Yang Xu, Tao Zhang, and Huansheng Ning. 2019. Review on smart gas sensing technology. *Sensors*, 19(17):3760.

Patrique Fiedler, Carlos Fonseca, Eko Supriyanto, Frank Zanow, and Jens Haueisen. 2022. A high-density 256-channel cap for dry electroencephalography. *Human brain mapping*, 43(4):1295–1308.

Marcus Georgi, Christoph Amma, and Tanja Schultz. 2015. Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing. In *International Conference on Bio-inspired Systems and Signal Processing*, volume 2, pages 99–108. Scitepress.

Ary L Goldberger, Zachary D Goldberger, and Alexei Shvilkin. 2017. *Clinical Electrocardiography: A Simplified Approach: Clinical Electrocardiography: A Simplified Approach E-Book*. Elsevier Health Sciences.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Sirui Hong, Yizhang Lin, Bangbang Liu, Binhao Wu, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Lingyao Zhang, Mingchen Zhuge, et al. 2024. Data interpreter: An llm agent for data science. *arXiv preprint arXiv:2402.18679*.

John D Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95.

Chihiro Ito, Xin Cao, Masaki Shuzo, and Eisaku Maeda. 2018. Application of cnn for human activity recognition with fft spectrogram of acceleration and gyro sensors. In *ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers*, pages 1503–1510.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023a. Timellm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, et al. 2023b. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv preprint arXiv:2310.10196*.

Yubin Kim, Xuhai Xu, Daniel McDuff, Cynthia Breazeal, and Hae Won Park. 2024. Health-llm: Large language models for health prediction via wearable sensor digital. In *Conference on Health, Inference, and Learning*, Proceedings of Machine Learning Research, pages 1–15. PMLR.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024a. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Li Liu, Yuxin Peng, Shu Wang, Ming Liu, and Zigang Huang. 2016. Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. *Information Sciences*, 340:41–57.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2024. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Dominique Makowski, Tam Pham, Zen J Lau, Jan C Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and SH Annabel Chen. 2021. Neurokit2: A python toolbox for neurophysiological signal processing. *Behavior research methods*, pages 1–8.

Suvir Mirchandani, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy Zeng. 2023. Large language models as general pattern machines. In *Conference on Robot Learning*, pages 2498–2518. PMLR.

OpenAI. 2022. Introducing chatgpt. https://www.openai.com/blog/chatgpt.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/.

Mehmet Akif Ozdemir, Deniz Hande Kisa, Onan Guren, and Aydin Akan. 2022. Dataset for multi-channel surface electromyography (semg) signals of hand gestures. *Data in brief*, 41:107921.

Alexandros Pantelopoulos and Nikolaos G Bourbakis. 2009. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 40:1–12.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in Neural Information Processing Systems*, 34:11054–11070.

Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. 2006. Feature selection and activity recognition from wearable sensors. In *Ubiquitous Computing Systems: Third International Symposium. Proceedings 3*, pages 516–527. Springer.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. 2018. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *ACM international conference on multimodal interaction*, pages 400–408.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Rajendra P Sishodia, Ram L Ray, and Sudhir K Singh. 2020. Applications of remote sensing in precision agriculture: A review. *Remote sensing*, 12(19):3136.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *ACM conference on embedded networked sensor systems*, pages 127–140.

Nils Strodthoff, Temesgen Mehari, Claudia Nagel, Philip J Aston, Ashish Sundar, Claus Graff, Jørgen K Kanters, Wilhelm Haverkamp, Olaf Dössel, Axel Loewe, et al. 2023. Ptb-xl+, a comprehensive electrocardiographic feature dataset. *Scientific data*, 10(1):279.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.

Yunus Emre Ustev, Ozlem Durmaz Incel, and Cem Ersoy. 2013. User, device and orientation independent human activity recognition on mobile phones: Challenges and a proposal. In *ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1427–1436.

Vini Vijayan, James P Connolly, Joan Condell, Nigel McKelvey, and Philip Gardiner. 2021. Review of wearable devices and data collection considerations for connected health. *Sensors*, 21(16):5589.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Gideon Vos, Kelly Trinh, Zoltan Sarnyai, and Mostafa Rahimi Azghadi. 2023. Generalizable machine learning for stress monitoring from wearable

devices: a systematic literature review. *International Journal of Medical Informatics*, 173:105026.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. 2020. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):1–15.

Yan Wang, Shuang Cang, and Hongnian Yu. 2019. A survey on wearable sensor modality centred human activity recognition in health care. *Expert Systems with Applications*, 137:167–190.

Chaoyi Wu, Jiayu Lei, Qiaoyu Zheng, Weike Zhao, Weixiong Lin, Xiaoman Zhang, Xiao Zhou, Ziheng Zhao, Ya Zhang, Yanfeng Wang, et al. 2023. Can gpt-4v (ision) serve medical applications? case studies on gpt-4v for multimodal medical diagnosis. *arXiv preprint arXiv:2310.09909*.

Dezhen Xiong, Daohui Zhang, Xingang Zhao, and Yiwen Zhao. 2021. Deep learning for emg-based human-machine interaction: A review. *Journal of Automatica Sinica*, 8(3):512–533.

Huatao Xu, Liying Han, Qirui Yang, Mo Li, and Mani Srivastava. 2024. Penetrative ai: Making llms comprehend the physical world. In *International Workshop on Mobile Computing Systems and Applications*, pages 1–7.

Hao Xue and Flora D Salim. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *Transactions on Knowledge and Data Engineering*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Han Yu, Peikun Guo, and Akane Sano. 2023. Zero-shot ecg diagnosis with large language models and retrieval-augmented generation. In *Machine Learning for Health (ML4H)*, pages 650–663. PMLR.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024a. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Extractive summarization via chatgpt for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024b. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. 2024c. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. 2024. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in Neural Information Processing Systems*, 36:43322–43355.

Table 3: Performance of text-only and visual prompts, both with and without using CoT. The highest accuracy values are noted in bold.

| | Accel. | | ECG | | |
Prompt	HHAR	Swim	PTB-XL (CD)	PTB-XL (MI)	Avg.
Text-only	0.66	0.51	0.73	0.62	0.63
Text-only (CoT)	0.51	0.25	0.63	0.53	0.48
Visual	**0.67**	**0.73**	**0.80**	0.68	0.72
Visual (CoT)	0.63	0.67	**0.80**	**0.73**	0.71

Table 4: Comparison of summarized text prompts with long text-only and our visual prompts. The highest accuracy values are noted in bold.

| | Accel. | | ECG | | |
Prompt	HHAR	Swim	PTB-XL (CD)	PTB-XL (MI)	Avg.
Summarized text	0.58	0.43	0.53	0.53	0.52
Text-only	0.66	0.51	0.73	0.62	0.63
Visual	**0.67**	**0.73**	**0.80**	**0.68**	0.72

A Effect of Zero-shot Chain-of-Thoughts

We experimented with zero-shot Chain-of-Thought (CoT) prompting (Kojima et al., 2022) by adding "let's think step-by-step" to our prompts, testing this on two accelerometers and two ECG datasets. Table 3 shows the findings. While CoT prompting is generally known to enhance LLM response quality, our results showed inconsistent performance by datasets. Notably, CoT consistently dropped performance for text-only prompts. We analyzed the results by observing the CoT responses, illustrated as examples in Figures 8 and Figure 9, showing wrong predictions with CoT from the HHAR dataset. We found that CoT reasoning in text-only prompts primarily focused on simple statistical comparisons, such as whether values were higher or lower. This simplistic approach proved inadequate for analyzing the complexities of sensor data, leading to suboptimal responses. Likewise, visual prompts indicated reasoning centered around terms like "variations," "periodic," and "stable," but they lacked the necessary depth to effectively assess more intricate features like frequency trends or signal shapes. This superficial reasoning suggests a significant gap in the CoT approach, underscoring the need for more task-specific reasoning prompts for sensory data analysis.

B Effect of Text Summarization

As discussed in Section 3, long numeric sequences in text increase costs and degrade performance. One way to address the challenge is through text summarization to reduce prompt length. However, no established method effectively summarizes sensor data, as different tasks require distinct features. A potential solution is prompting-based summarization (Zhang et al., 2023, 2024b) that instructs MLLMs to extract key information using a general prompt: "summarize the given text." To explore this, we prompted GPT-4o to "summarize the

pattern or tendency of the data" aiming to reduce prompt length in text. We specified the focus on patterns and tendencies to allow for fair comparison, as our visualizations typically capture these aspects. This method was tested on two accelerometer and two ECG datasets.

Table 4 presents the results. Although summarized text showed feasibility over random predictions, it underperformed compared to both text-only and visual prompts. This highlights the challenge of summarizing sensor data effectively in text. The results suggest that exploring generalizable approaches to reduce text for sensory tasks remains an future research.

C Small MLLMs on Sensory Tasks

We used GPT-4o, the latest and most accessible model supporting both vision and text inputs. To test the generalizability of our approach on smaller MLLMs, we conducted experiments on four datasets using LLaVa-7B (Liu et al., 2024a). For the test, we used the interleaved version, which allows multi-image input to enable few-shot prompting. Due to memory limitations, LLaVa-7B could not handle text-only prompts with large tokens. For instance, prompts with more than 50K tokens from the HHAR dataset required over 150GB of VRAM per inference, making it infeasible. We evaluated text-only prompts of ECG datasets, which contained fewer than 5K tokens.

As shown in Table 5, LLaVa-7B performed poorly on sensory tasks, yielding results close to random predictions for both text-only and visual prompts. We believe this is due to the model's smaller size and lack of pre-training, limiting its ability to interpret complex graphs, plots, and data patterns. Its limited capacity for multi-image understanding (Zhao et al., 2024) may also have affected its analysis of the provided examples. Future research should focus on enhancing low-capacity models for sensory data tasks.

Table 5: Performance comparison of text-only and visual prompts using LLaVa-7B and GPT-4o as the MLLMs. Accelerometer text-only prompts were not evaluated due to excessive VRAM consumption over 150GB during local inference with LLaVa. The highest accuracy values are highlighted in bold.

| | Accel. | | ECG | | |
Prompt	HHAR	Swim	PTB-XL (CD)	PTB-XL (MI)	Avg.
LLaVa-7B					
Text-only	-	-	0.50	0.48	-
Visual	0.15	0.20	0.50	0.48	0.33
GPT-4o					
Text-only	0.66	0.51	0.73	0.62	0.63
Visual	**0.67**	**0.73**	**0.80**	**0.68**	0.72

D Use of Subplots for Multi-channel Data

Sensor data often include multiple channels. Our visual prompts differentiated channels using varying colors within a single plot to maintain a shared axis system. To assess the impact of different plotting approaches, we conducted experiments using accelerometer datasets, which have three channels. Specifically, we compared visualizing three distinct plots for each channel against our current approach. Table 6 shows the results. The results indicated that separated plots for each channel reduced performance by 12%. We hypothesize that multiple subplots distribute visual features over different regions, resulting in problems in understanding the relationship between different channels. To this end, we recommend using an aggregated plot when all channels can be represented within a plot. However, for dense datasets, such as 256-channel EEG (Fiedler et al., 2022), a single plot may not suffice, highlighting a limitation in our current visualization approach. Addressing this challenge will be a focus of future research.

E Visualization Tools

Our visualization generator employs tools available in public libraries to create visualizations. We have equipped the visualization generator with 16 distinct visualization functions sourced from widely used libraries such as Matplotlib (Hunter, 2007), Scipy (Virtanen et al., 2020), and Neurokit2 (Makowski et al., 2021). The specific visualization tools implemented in our generator and their descriptions are outlined in Table 7. The descriptions presented in the table were directly written inside the prompt for the visualization tool filtering (see Appendix G).

Table 6: Performance comparison of visualizing multi-channel sensor data (accelerometer) using a single plot versus multiple subplots. The single plot method combines multiple waveforms in one shared-axis plot, each channel distinguished by color coding.

Plotting approach	HHAR	UTD-MHAD	Swim	Avg.
Single plot	**0.67**	**0.43**	**0.74**	**0.61**
Multiple subplots	0.53	0.31	0.69	0.51

F Details of Sensory Tasks

We conducted experiments across nine sensory tasks across four sensor modalities, each with unique objectives. This section provides the details of these tasks, including task descriptions, classifications, sampling rates, window durations, and data collection protocols. We directly followed the given sampling rate with the original dataset to represent data in text prompts. The descriptions of each dataset are used to formulate the instructions for our visual prompts. The complete prompt examples are in Appendix G.

Human activity recognition: We used the HHAR (Stisen et al., 2015) dataset to classify six basic human activities: sit, stand, walk, bike, upstairs, and downstairs. Data were collected from the built-in accelerometers of smartphones and smartwatches along with the x, y, and z axes. Due to strong domain effects (Ustev et al., 2013), we exclusively used smartwatch data for the experiment. The data, sampled at 100Hz, were segmented into 5-second windows following the established practice for human activity recognition (Altun and Barshan, 2010).

Complex activity recognition: We used the UTD-MHAD (Chen et al., 2015) dataset to classify a wide array of 21 activities: swipe left, swipe right, wave, clap, throw, arms cross, basketball shoot, draw X, draw a circle (clockwise), draw a circle (counter-clockwise), draw a triangle, bowling, boxing, baseball swing, tennis swing, arm curl, tennis serve, push, knock, catch, and pickup and throw. Accelerometers attached to the users' right wrist were used for data collection. We used data sampled at 50Hz with 3-second windows as described in the dataset documentation.

Swimming style recognition: The swimming dataset (Brunner et al., 2019) involves acceleration data from swimmers performing five different styles: backstroke, breaststroke, butterfly, freestyle,

To determine the most likely activity from the given sensor data, we need to analyze the patterns and characteristics of the data along the x, y, and z axes. Let's break down the data step-by-step and compare it with the provided examples for each activity.

Analysis of the Given Data
1. **X-axis**: The values range from approximately -2.94 to 0.79.
2. **Y-axis**: The values range from approximately 1.3 to 2.71.
3. **Z-axis**: The values range from approximately -1.42 to 0.16.

Comparison with Examples
1. **Bike**:
- The bike data shows significant variations in all three axes, with values often exceeding 1.0 or -1.0.
- The given data does not show such high variations, especially in the x-axis and z-axis.

2. **Sit**:
- The sit data shows relatively stable values, especially in the y-axis, which is consistently around -0.8.
- The given data has a y-axis range from 1.3 to 2.71, which is not consistent with sitting.

3. **Stand**:
- The stand data shows relatively stable values, especially in the y-axis, which is consistently around -0.3.
- The given data has a y-axis range from 1.3 to 2.71, which is not consistent with standing.

4. **Walk**:
- The walk data shows periodic variations in all three axes, with values often exceeding 1.0 or -1.0.
- The given data does not show such periodic variations, especially in the x-axis and z-axis.

5. **Stairsdown**:
- The stairsdown data shows significant variations in all three axes, with values often exceeding 1.0 or -1.0.
- The given data does not show such high variations, especially in the x-axis and z-axis.

6. **Stairsup**:
- The stairsup data shows significant variations in all three axes, with values often exceeding 1.0 or -1.0.
- The given data does not show such high variations, especially in the x-axis and z-axis.

Conclusion
The given data shows relatively stable values in the x-axis and z-axis, with the y-axis values consistently high (around 1.3 to 2.71). This pattern is most consistent with the activity of **sit**, where the y-axis values are stable but not as high as in the given data. However, considering the options and the characteristics of the data, the most likely activity is **sit**.

<answer>sit</answer>

Figure 8: An example CoT response from a text-only prompt designed for the HHAR task. The correct prediction is "walk", while the MLLM outputs "sit."

Let's analyze the target data step-by-step by comparing it with the provided examples:

1. **Sit**: The data for sitting shows very little variation in all three axes, with values remaining close to zero. The target data shows significant variation, so it is not sitting.

2. **Stand**: The data for standing shows some variation, but it is relatively stable compared to other activities. The target data shows more variation than the standing data, so it is unlikely to be standing.

3. **Walk**: The data for walking shows periodic variations in all three axes. The target data does not show such periodic patterns, so it is unlikely to be walking.

4. **Stairsup**: The data for walking upstairs shows periodic variations similar to walking but with a different pattern. The target data does not match this pattern.

5. **Stairsdown**: The data for walking downstairs also shows periodic variations. The target data does not match this pattern.

6. **Bike**: The data for biking shows variations in all three axes, but it is less periodic and more irregular compared to walking or stairs activities. The target data shows similar irregular variations.

Based on the comparison, the target data most closely resembles the data for biking.

<answer>bike</answer>

Figure 9: An example CoT response from a visual prompt designed for the HHAR task. The correct prediction is "stand", while the MLLM outputs "bike."

and stationary. This dataset evaluates performance in sports-specific contexts. Data were collected from wrist-worn accelerometers and sampled at 30Hz. We used the 3-second windows recommended with the dataset.

Four arrhythmia detections: The PTB-XL (Wagner et al., 2020) dataset contains ECG recordings from patients with four different types: Conduction Disturbance (CD), Myocardial Infarction (MI), Hypertrophy (HYP), and ST/T Change (STTC). We defined each type as a binary classification task. The dataset comprises 10-second records from clinical 12-lead sensors sampled at 100Hz. We used lead II, the most commonly used lead for arrhythmia detection (Goldberger et al., 2017).

Hand gesture recognition: We included a dataset (Ozdemir et al., 2022) classifying ten different hand gestures using EMG signals: rest, extension, flexion, ulnar deviation, radial deviation, grip, abduction of fingers, adduction of fingers, supination, and pronation. Data were collected from four forearm surface EMG sensors with a 2000Hz sampling rate. We utilized all four channels with a 0.2-second window, following an existing practice known to be effective (Georgi et al., 2015).

Stress Detection: The WESAD (Schmidt et al., 2018) dataset is designed for stress detection (baseline, stress, amusement) from multiple wearable sensors. We focused exclusively on respiration data measured from the chest for a distinct evaluation setting. The sensor was attached to the users' chests, with data collected at 700Hz. Following the official guidelines, we employed the three-class classification task (baseline, stress, amusement) using 10-second windows.

G Prompts

We present examples of prompts used in our experiments. Figure 10 and Figure 11 illustrate two text-only prompt examples derived from the HHAR and PTB-XL (CD) datasets; in these examples, sensor data is truncated after a certain point to conserve space, though the format remains consistent with varying values. Figure 12 and Figure 13 displays the visual prompts created for the same datasets, HHAR and PTB-XL (CD). Figure 14 details the prompt for our visualization tool filtering specific to the PTB-XL (CD) task, with demonstrations omitted and presented separately in Figure 15. Lastly, Figure 16 showcases the visualization selection prompt for the PTB-XL (CD) dataset.

Instruction
You are an expert in sensor data analysis. Given the sensor data, determine the correct answer from the options listed in the question. Provide the answer with the format of <answer>ANSWER</answer>, where ANSWER corresponds to one of the options listed in the question. If the answer is not in the options, choose the most possible option.

The sensor data is collected from an accelerometer measuring acceleration along the x, y, and z axes. The data is normalized with the statistics of the user's data. The data is collected over 5 seconds. The data is measured from a smartwatch which was attached to the wrist of a user. Please refer to the provided examples and use them to answer the following question for the target data.

Examples
Example of stand:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[-0.75, 0.47, -1.27], [-0.75, 0.51, -1.28], [-0.75, 0.49, -1.26], …

Example of sit:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[0.78, -1.36, 1.11], [0.78, -1.37, 1.11], [0.78, -1.37, 1.11], ..

Example of walk:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[-0.45, 0.44, -1.01], [-0.5, 0.6, -0.99], [-0.39, 0.54, -0.98], …

Example of stairsup:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[2.87, 0.65, -3.29], [2.75, 0.77, -3.43], [2.59, 0.86, -3.39], …

Example of stairsdown:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[-0.96, -0.32, 0.38], [-1.25, 0.02, 0.74], [-1.26, 0.09, 0.58], …

Example of bike:
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[-0.44, 0.67, 0.59], [-0.64, 1.03, 0.97], [-0.84, 1.68, 0.05], …

Question
Given sensor data (list of ['X-axis', 'Y-axis', 'Z-axis']): [[-0.38, 0.43, -0.88], [-0.16, 0.63, -0.91], [-0.19, 0.56, -0.91], …

Question: When the sensor data is used for a task for classifying 6 human activities, bike, sit, stand, walk, stairsdown, stairsup, using three-axis accelerometer data measured from a wrist-worn smartwatch, what is the most likely answer among ['bike', 'sit', 'stand', 'walk', 'stairsdown', 'stairsup']?
Answer:

Figure 10: An example of a text-only prompt for solving the HHAR task. The sensor data represented in the text are truncated beyond a certain point.

Instruction
You are an expert in sensor data analysis. Given the sensor data, determine the correct answer from the options listed in the question. Provide the answer with the format of <answer>ANSWER</answer>, where ANSWER corresponds to one of the options listed in the question. If the answer is not in the options, choose the most possible option.

The ECG data is collected from a lead II ECG sensor. The ECG data is recorded over 10 seconds. The data is normalized with the statistics of the user's data. Please refer to the provided examples and use them to answer the following question for the target data.

Examples
Example of normal:
Average heartbeat in the ECG signal (list of ['lead II']): [-0.34, -0.34, -0.35, -0.35, -0.36, …
ECG_P_Peaks in the ECG signal (list of (index, value)): [(21, 0.08), (129, 0.19), (239, 0.22), …
ECG_Q_Peaks in the ECG signal (list of (index, value)): [(35, -0.61), (137, -0.46), (246, -0.48), …
ECG_S_Peaks in the ECG signal (list of (index, value)): [(42, -1.4), (149, -1.35), (260, -1.33), …
ECG_T_Peaks in the ECG signal (list of (index, value)): [(63, 2.18), (171, 2.11), (282, 2.31), …

Example of conduction disturbance:
Average heartbeat in the ECG signal (list of ['lead II']): [-0.15, -0.24, -0.29, -0.31, -0.28, …
ECG_P_Peaks in the ECG signal (list of (index, value)): [(4, 0.14), (57, 0.3), (103, 0.22), …
ECG_Q_Peaks in the ECG signal (list of (index, value)): [(14, -0.05), (65, -0.05), (109, -0.07), …
ECG_S_Peaks in the ECG signal (list of (index, value)): [(22, -2.28), (73, -2.1), (124, -2.35), …
ECG_T_Peaks in the ECG signal (list of (index, value)): [(82, 0.03), (142, 0.64), (245, 0.25), …

Question
Average heartbeat in the ECG signal (list of ['lead II']): [-0.39, -0.39, -0.39, -0.39, -0.4, …
ECG_P_Peaks in the ECG signal (list of (index, value)): [(15, 0.14), (94, -0.26), (173, -0.23), …
ECG_Q_Peaks in the ECG signal (list of (index, value)): [(23, -0.27), (102, -0.81), (182, -0.55), …
ECG_S_Peaks in the ECG signal (list of (index, value)): [(34, 0.15), (116, -0.51), (192, -0.45), …
ECG_T_Peaks in the ECG signal (list of (index, value)): [(50, 1.39), (130, 1.1), (209, 1.31), …

Question: When the sensor data is used for a task for classifying ECG data into 2 categories: conduction disturbance, normal, what is the most likely answer among ['conduction disturbance', 'normal']?
Answer:

Figure 11: An example of a text-only prompt for solving the PTB-XL (CD) task. The sensor data represented in the text are truncated beyond a certain point.

Instruction
You are an expert in sensor data analysis. Given the sensor data, determine the correct answer from the options listed in the question. Provide the answer with the format of <answer>ANSWER</answer>, where ANSWER corresponds to one of the options listed in the question. If the answer is not in the options, choose the most possible option.

The sensor data is collected from an accelerometer measuring acceleration along the x, y, and z axes. The data is normalized with the statistics of the user's data.The data is collected over 5 seconds. The data is measured from a smartwatch which was attached to the wrist of a user. Please refer to the examples provided in the images and use them to answer the following question for the target data.

Question
Question: When the sensor data is used for a task for classifying 6 human activities, bike, sit, stand, walk, stairsdown, stairsup, using three-axis accelerometer data measured from a wrist-worn smartwatch, what is the most likely answer among ['bike', 'sit', 'stand', 'walk', 'stairsdown', 'stairsup']?
Answer:

Figure 12: An example of a visual prompt for solving the HHAR task.

Instruction
You are an expert in sensor data analysis. Given the sensor data, determine the correct answer from the options listed in the question. Provide the answer with the format of <answer>ANSWER</answer>, where ANSWER corresponds to one of the options listed in the question. If the answer is not in the options, choose the most possible option.

The ECG data is collected from a lead II ECG sensor. The ECG data is recorded over 10 seconds. The data is normalized with the statistics of the user's data. Please refer to the examples provided in the images and use them to answer the following question for the target data.

Question
Question: When the sensor data is used for a task for classifying ECG data into 2 categories: conduction disturbance, normal, what is the most likely answer among ['conduction disturbance', 'normal']?
Answer:

Figure 13: An example of a visual prompt for solving the PTB-XL (CD) task.

Instructions
You need to determine effective visualization methods for the given task. Provide visualization methods that aid in analyzing the data for this task, along with the required arguments for that method. Additionally, explain how to use the information from the visualization to solve the task. You can provide several candidates as a list. Generate the answer in the following format:
[{"func": visualization_method, "args": {"arg1": arg1_val, "arg2": arg2_val, ...}], "knowledge": knowledge}, ...]

The available visualization methods are as follows:

raw waveform: This generates a raw signal of sensor data, displaying the amplitude of the signal over time. This is usually used to visualize the raw data and identify patterns in the signal.
spectrogram: This generates a spectrogram of sensor data, …

Demonstrations
Data description: The sensor data is collected from an accelerometer measuring acceleration along the x, y, and z axes. The data is normalized with the statistics of the user's data. The data is measured from an accelerometer attached to the ankles of a user.
Task description: A task for classifying running and walking activities using accelerometer data measured from an ankle-worn device.
Response: {"func": "raw waveform", "args": {}, "knowledge": "Use this to visualize the amplitude of the accelerometer signal over time. For classifying running and walking, observe the patterns in the waveform: running typically shows higher amplitude and more frequent peaks due to the higher impact and faster motion, while walking shows lower amplitude and less frequent peaks."}
…

Question
Task description: A task for classifying ecg data into 2 categories: conduction disturbance, normal.
Data description: The ecg data is collected from a lead ii ecg sensor. the ecg data is recorded over 10 seconds. the data is normalized with the statistics of the user's data.
Response:

Figure 14: An example prompt from our visualization generator for visualization tool filtering in the PTB-XL (CD) task. Demonstrations are omitted in this example but can be found in Figure 15.

Demonstrations
Data description: The sensor data is collected from an accelerometer measuring acceleration along the x, y, and z axes. The data is normalized with the statistics of the user's data. The data is measured from an accelerometer attached to the ankles of a user.
Task description: A task for classifying running and walking activities using accelerometer data measured from an ankle-worn device.
Response: {"func": "raw waveform", "args": {}, "knowledge": "Use this to visualize the amplitude of the accelerometer signal over time. For classifying running and walking, observe the patterns in the waveform: running typically shows higher amplitude and more frequent peaks due to the higher impact and faster motion, while walking shows lower amplitude and less frequent peaks."}

Data description: The sensor data is collected from an accelerometer measuring acceleration along the x, y, and z axes. The data is normalized with the statistics of the vehicle's data. The data is measured from an accelerometer attached to a vehicle.
Task description: A task for classifying road types, such as asphalt, dirt, and cobblestone, using accelerometer data measured from a vehicle.
Response: {"func": "spectrogram", "args": {"nfft": 128, "nperseg": 128, "noverlap": 120, "mode": "magnitude"], "knowledge": "Use this to analyze the frequency components of the accelerometer signal over time. The colors in the spectrogram represent the magnitude of the frequencies: brighter colors indicate higher magnitudes. For road type classification, asphalt typically shows lower frequency components with smoother patterns, dirt shows higher frequency components with irregular patterns, and cobblestone shows high-frequency components with periodic patterns due to the regular bumps."}

Data description: The sensor data is collected from an ECG measuring the electrical activity of the heart. The data is normalized with the statistics of the user's data. The data is measured using electrodes attached to the chest of a user.
Task description: A task for detecting sleep apnea using ECG data measured from chest electrodes.
Response: {"func": "ECG individual heart beats", "args": {}, "knowledge": "Use this to aggregate and visualize individual heartbeats within an ECG recording. In normal beats, the P-wave precedes the QRS complex, and the T-wave follows it. In sleep apnea, irregularities in the intervals between the P, Q, R, S, and T peaks can be observed. For instance, the absence of regular QRS complexes or prolonged intervals can indicate episodes of apnea. The plot helps identify these patterns by showing the average shape of the heartbeats and marking the specific peaks."}

Data description: The sensor data is collected from an EMG sensor measuring muscle electrical activity. The data is normalized with the statistics of the user's data. The data is measured using electrodes attached to the forearm of a user.
Task description: A task for recognizing finger gestures, such as numbers, using EMG data measured from forearm electrodes.
Response: {"func": "EMG signal", "args": {}, "knowledge": "Use this to visualize the raw EMG signal over time. For recognizing finger gestures, observe the patterns and amplitude of muscle activity. Different numbers (gestures) will produce distinct patterns in the EMG signal. For example, bending more fingers usually results in higher amplitude signals due to increased muscle activation."}

Data description: The sensor data is collected from an ECG measuring the electrical activity of the heart. The data is normalized with the statistics of the user's data. The data is measured using electrodes attached to the chest of a user.
Task description: A task for detecting whether the user is running or not using ECG data measured from chest electrodes.
Response: {"func": "ECG heart rate", "args": {}, "knowledge": "Use this to monitor heart rate over time and analyze activity levels. A significant increase in heart rate can indicate that the user is running. The plot should show a higher average heart rate during running periods compared to resting or walking periods. Sudden spikes and consistent high heart rates are typical indicators of running."}

Figure 15: Demonstrations provided inside the visualization tool filtering prompt to enhance the response quality.

Instruction
You do not have any prior knowledge about sensor data and visualization techniques.
Based solely on the visual cues in the provided images,
identify the visualization that most visually distinguishes all different classes for the given task.
Generate the answer in the following format:
{"func": visualization_method}

Question
Visualization methods: ['ECG individual heart beats', 'ECG signal and peaks', 'ECG heart rate']
Task description: A task for classifying ecg data into 2 categories: conduction disturbance, normal.
Data description: The ecg data is collected from a lead ii ecg sensor. the ecg data is recorded over 10 seconds. the data is normalized with the statistics of the user's data.
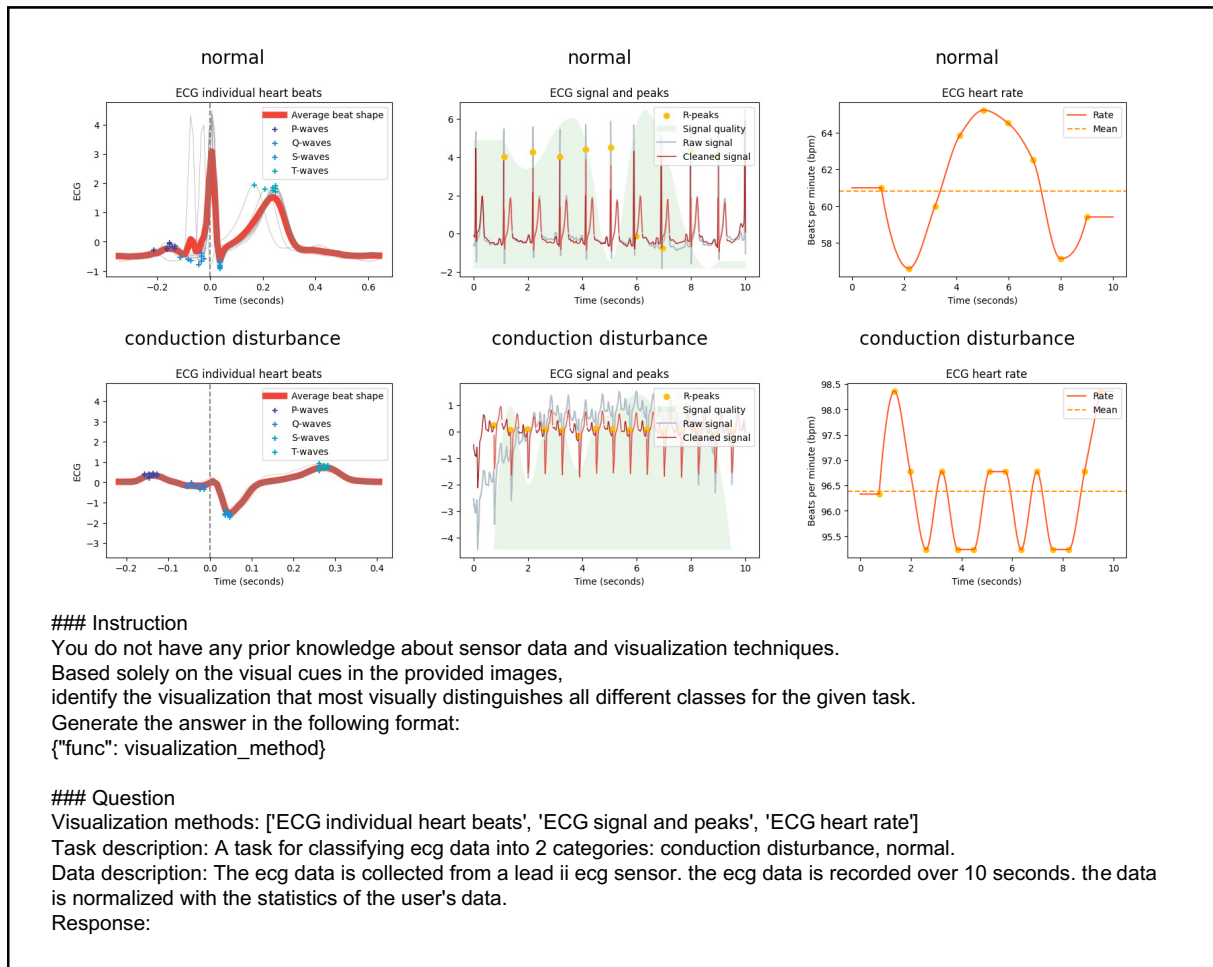Response:

Figure 16: An example prompt from our visualization generator for visualization selection in the PTB-XL (CD) task.

Table 7: Descriptions of the visualization tools provided to our visualization generator.

Visualization tool	Description
raw waveform	This generates a raw signal of sensor data, displaying the amplitude of the signal over time. This is usually used to visualize the raw data and identify patterns in the signal.
spectrogram	This generates a spectrogram of sensor data, showing the density of frequencies over time. This is usually used to visualize the frequency components for high-frequency data, which has features over components but is hard to figure out in the raw plot. It takes the length of the FFT used (nfft), the length of each segment (nperseg), and the number of points to overlap between segments (noverlap) as parameters. Different modes (mode) can be defined to specify the type of return values: ["psd" for power spectral density, "complex" for complex-valued STFT results, "magnitude" for absolute magnitude, "angle" for complex angle, and "phase" for unwrapped phase angle]. (Arguments: nfft, nperseg, noverlap, mode)
signal power spectrum density	This generates a power spectrum density plot, which shows the power of each frequency component of the signal on the x-axis. This is usually used to analyze the signal's power distribution of different frequency components.
EDA signal	This generates a plot showing both raw and cleaned Electrodermal Activity (EDA) signals over time. This is usually used to analyze the EDA signals for patterns related to stress, arousal, or other psychological states.
EDA skin conductance response (SCR)	This generates a plot of skin conductance response (SCR) for EDA data, highlighting the phasic component, onsets, peaks, and half-recovery times. This is usually used to study the transient responses in EDA data related to specific stimuli or events.
EDA skin conductance level (SCL)	This generates a plot of skin conductance level (SCL) for EDA data over time. This is usually used to analyze the tonic component of EDA data, reflecting the overall level of arousal or stress over a period.
ECG signal and peaks	This generates a plot for Electrocardiogram (ECG) data, showing the raw signal, cleaned signal, and R peaks marked as dots to indicate heartbeats. This is usually used to analyze the heartbeats and detect anomalies in the ECG signal.
ECG heart rate	This generates a heart rate plot for ECG data, displaying the heart rate over time and its mean value. This is usually used to monitor and analyze heart rate variability and trends over time.
ECG individual heartbeats	This generates a plot of individual heartbeats and the average heart rate for ECG data. It aggregates heartbeats within an ECG recording and shows the average beat shape, marking P-waves, Q-waves, S-waves, and T-waves. This is usually used to study the morphology of individual heartbeats and identify irregularities.
PPG signal and peaks	This generates a plot for Photoplethysmogram (PPG) data, showing the raw signal, cleaned signal, and systolic peaks marked as dots. This is usually used to analyze the blood volume pulse and detect anomalies in the PPG signal.
PPG heart rate	This generates a heart rate plot for PPG data, displaying the heart rate over time and its mean value. This is usually used for PPG data to monitor and analyze heart rate variability and trends over time.
PPG individual heartbeats	This generates a plot of individual heartbeats and the average heart rate for PPG data, aggregating individual heartbeats within a PPG recording and showing the average beat shape. This is usually used to study the morphology of individual heartbeats based on PPG data.
EMG signal	This generates a plot showing both raw and cleaned Electromyogram (EMG) signals over time. This is usually used to analyze muscle activity and identify patterns in muscle contractions.
EMG muscle activation	This generates a muscle activation plot for EMG data, displaying the amplitudes of muscle activity and highlighting activated parts with lines. This is usually used to study muscle activation levels and identify specific periods of muscle activity.
EOG signal	This generates a plot showing both raw and cleaned Electrooculogram (EOG) signals over time, with blinks marked as dots. This is usually used to analyze eye movement patterns and detect blinks.
EOG blink rate	This generates a blink rate plot for EOG data, displaying the blink rate over time and its mean value. This is usually used to monitor and analyze the blink rate and detect irregularities.
EOG individual blinks	This generates a plot of individual blinks for EOG data, aggregating individual blinks within an EOG recording and showing the median blink shape. This is usually used to study the morphology of individual blinks and identify patterns in blink dynamics.