Evaluating Visual Prompts with Eye-Tracking Data for MLLM–Based Human Activity Recognition

Jae Young Choi[*] KAIST
Seon Gyeom Kim KAIST
Hyungjun Yoon KAIST
Taeckyung Lee KAIST
Donggun Lee KAIST
Jaeryung Chung KAIST
Jihyung Kil Adobe Research
Ryan Rossi Adobe Research
Sung-Ju Lee KAIST
Tak Yeon Lee[†] KAIST

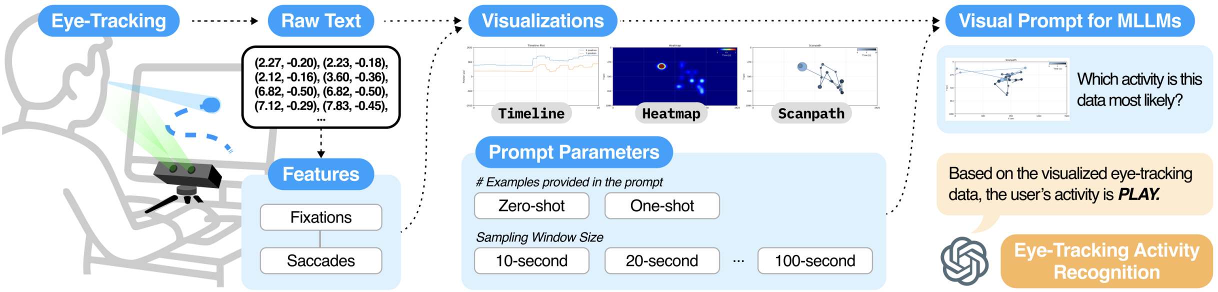

Figure 1: Overview of the study. We systematically explored visual prompting strategies with eye-tracking data by varying visualization types and prompt parameters (e.g., zero/one-shot and windowing sizes) for MLLM-based human activity recognition.

Abstract

Large Language Models (LLMs) have emerged as foundation models for IoT applications such as human activity recognition (HAR). However, directly applying high-frequency and multi-dimensional sensor data, such as eye-tracking data, leads to information loss and high token costs. To mitigate this, we investigate a visual prompting strategy that transforms sensor signals into data visualization images as an input to multimodal LLMs (MLLMs) using eye-tracking data. We conducted a systematic evaluation of MLLM-based HAR across three public eye-tracking datasets using three visualization types of timeline, heatmap, and scanpath, under varying temporal window sizes. Our findings suggest that visual prompting provides a token-efficient and scalable representation for eye-tracking data, highlighting its potential to enable MLLMs to effectively reason over high-frequency sensor signals in IoT contexts.

Index Terms: Ubiquitous Computing; Human Activity Recognition; Multimodal LLM; Foundation Model; Eye-Tracking Visualization; Visual Prompting

1 Introduction

In recent years, Large Language Models (LLMs) have been increasingly adopted as foundation models for IoT applications to understand human context and make autonomous decisions. Prior works have demonstrated grounding LLMs with diverse sensor data to support a wide range of IoT tasks, including human activity recognition (HAR) [1, 2], data sensemaking [3], and health-related state inference from wearable sensing [4]. More recently, LLMs have been explored as reasoning agents for IoT scenarios [5]. Collectively, these studies investigate and validate the feasibility of adopting LLMs as the foundation model for IoT ecosystems. Unlike traditional machine learning approaches, integration of LLMs enables training-free IoT applications, bypassing the laborious effort for massive labeled data collection and provides generalization capabilities with the LLM's flexible world knowledge [6].

However, significant challenges remain in utilizing LLMs in tasks with numeric, high-frequency time-series sensor data. First, LLM tokenizers are inherently designed for natural language and fail to capture patterns in continuous numerical sequences, resulting in the loss of temporal correlations [7]. Second, high-frequency raw sensor data expands to an extremely large number of tokens, incurring infeasible computational cost and often prohibitive token costs. Importantly, the lengthy input triggers the "lost in the middle" problem [8], causing critical temporal patterns embedded in long sensor data to be overlooked. To mitigate this, prior approaches have focused on aligning time-series modalities with LLMs, either by developing specialized tokenizers and encoders to handle numerical sequences efficiently [9] or by pre-training foundation models directly on large-scale time-series datasets such as inertial measurement unit (IMU) data [10, 11]. In particular, Yoon et al. [12] utilized visual prompting[1] to mitigate the problem, which transformed sensor data (e.g., accelerometer and physiological data) into visualizations such as line plots or spectrograms to be interpreted by MLLM. Their findings demonstrated that this visual prompting approach outperformed text-based prompting about 10% in average.

Meanwhile, eye-tracking serves as an aspiring sensing modality in intelligent systems, enabling diverse applications to infer user attention or cognition in pervasive environments such as smart homes and head-mounted displays [13, 14]. Despite its utility, effectively grounding LLMs in eye-tracking data remains difficult due to its high-frequency, spatio-temporal nature, which requires combining 2D spatial coordinates with temporal dynamics. At the same time, this characteristic allows diverse visualization techniques such as scanpaths [15] or heatmaps [16], which have been employed depending on specific analytical objectives. In this work, we leverage these distinctive visual patterns inherent in eye-tracking visualizations to examine their effectiveness as visual prompts for multimodal LLMs. To validate this approach, we employ HAR, one of the key tasks in LLM-based IoT research [1, 2]. Through this experiment, we aim to demonstrate how different visual prompting strategies, such as visualization techniques and window sizes, can effectively represent the distinct behavioral patterns captured across multiple eye-tracking datasets. Furthermore, we discuss the future development and broader applicability of this approach.

[*] e-mail: jaeyoungchoi@kaist.ac.kr
[†] e-mail: takyeonlee@kaist.ac.kr

[1] In this paper, we use the term *visual prompting* to denote the use of data visualization images as model inputs to guide a model with visual cues [12].

2 Related Works

2.1 LLM-Based Human Activity Recognition

HAR has long been a central component in IoT and Cyber-Physical Systems (CPS), as it enables systems to interpret and respond to human behaviors from sensor data. Traditional approaches have focused on developing task-specific models using rule-based, machine learning, or neural networks trained on labeled datasets. More recently, a growing body of research has explored the use of LLMs as foundation models for HAR [2, 6, 17, 18, 19, 20], motivated by reduced training costs and potential generalizability [21]. One of the major examples is HARGPT [2], which demonstrated zero-shot HAR by feeding IMU data into the model, leveraging its internal world knowledge. A recent study also explored LLMs' potential in fine-grained hand gesture recognition, highlighting the importance of domain-specific adaptation and few-shot learning [19].

In LLM-based HAR tasks, sensor data are commonly provided directly within the in-context prompt without additional model training. Prompts often employ a specific expert persona (e.g., "You are an expert in human activity analysis"), provide data collection context such as device specifications and sampling rates, and include few-shot examples when necessary. The prompt also explicitly specifies the target task (e.g., identifying the activity performed based on the provided data). When supplying time-series sensor data to LLMs via their context windows, the data are formatted in various ways, ranging from raw numerical strings [2, 18], statistical summaries [18, 20], to high-level natural language descriptions of signal trends [17, 20]. Among these diverse input strategies, our work adopts the visual prompting approach of Yoon et al. [12], transforming sensor data into images to leverage the visual reasoning capabilities of MLLMs. Specifically, we apply this approach to eye-tracking data, which is two-dimensional and temporal data.

2.2 Eye-Tracking Visualization Techniques

Eye-tracking data can be visualized in different ways depending on the analytical purpose and the type of information to be revealed. According to the taxonomy proposed by Blascheck et al. [22], eye-tracking visualizations can be categorized into (i) Area-of-Interest (AOI)-based or (ii) point-based. AOI-based visualizations focus on semantic information, analyzing the transition between predefined regions or objects [23]. In contrast, point-based visualizations utilize the spatial and temporal information of recorded data points (e.g., (x, y) coordinates) without requiring semantic annotations. Due to the unique nature of eye-tracking data, many point-based visualizations rely on preprocessed eye-movement features, most notably fixations and saccades. Fixations refer to relatively stable gaze periods lasting approximately 200–300 ms, whereas saccades denote rapid eye movements between fixations [24]. Feature extraction is commonly performed using velocity-based or dispersion-based methods, including I-VT, I-DT, and I-HMM [25].

Within point-based visualizations, techniques can be differentiated based on their temporal and spatial emphasis. **Timeline** plot represents temporal characteristics of gaze data by mapping gaze positions or derived features along a time axis. Typically, raw gaze coordinates are plotted as functions of time, enabling analysts to inspect temporal fluctuations in gaze behavior [26]. More advanced variants additionally visualize fixations and saccades over time to explicitly encode attentional shifts [26, 27]. **Heatmap** represents the spatial distribution of gaze [22] by overlaying gaze positions directly onto the stimulus to convey where viewers focused their attention [16]. As reported by Bojko [28], heatmaps can be further classified based on the level of aggregation, including representations of absolute raw gaze points and fixation-based heatmaps that encode fixation counts or durations. **Scanpath** depicts gaze trajectories to represent the spatio-temporal sequence of visual exploration [15]. A common usage involves rendering fixations as nodes and saccades as connecting paths, where variables such as the circle radius are often mapped according to the fixation duration [29]. Additional visual encoding, such as gradient, can be employed to convey the temporal order of sequences [30]. **Space–time cube (STC)** extends the 2D spatial domain of a stimulus with a temporal dimension, resulting in a 3D representation of gaze [31].

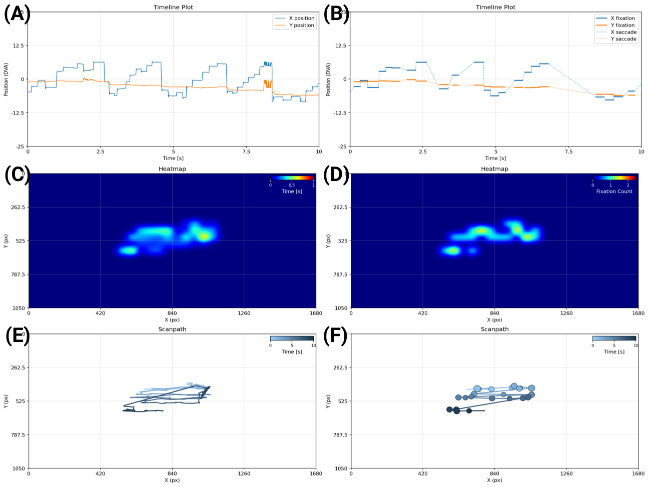

Figure 2: Visualizations used in this study. (A) Raw Timeline (B) Feature-based Timeline (C) Absolute Duration Heatmap (D) Fixation Count Heatmap (E) Raw Scanpath (F) Feature-based Scanpath.

Building upon these established visualization techniques, we investigate their use as visual prompts, whether the techniques originally designed for human analysis can also support visual reasoning in MLLMs. Compared to our prior work [32], we broaden the scope by considering a more diverse set of visualization techniques and varying temporal window sizes across multiple datasets.

3 Research Methodology

3.1 Visualization Selection and Design

In this study, we focus only on point-based eye-tracking visualization to accommodate general IoT scenarios where semantic AOIs are not readily defined. Specifically, we consider only 2D visualization techniques, excluding STCs. To systematically evaluate the efficacy of visual prompting across different visualization techniques, we designed six distinct visualizations (Figure 2), categorized into three primary representations: (i) **Timeline**, (ii) **Heatmap**, and (iii) **Scanpath**. For each category, we compared raw data representations with feature-based representations derived from fixations and saccades extracted using the I-DT algorithm [25]. While the dispersion thresholds were adaptively adjusted for each dataset, the minimum dwell time for fixation detection was fixed at 200 ms.

Regarding the visualization techniques, the Timeline plot using the *raw data* (Figure 2A) represents x and y coordinates as distinct colored lines over time. For the *feature-based* Timeline plot (Figure 2B), fixations are illustrated with thick solid lines, while saccades are depicted as dashed lines to distinguish gaze events. The Heatmaps (Figure 2C, D) were generated on a blue background by partitioning the screen into 50 px grids and using matplotlib's Gaussian interpolation for visual smoothing. Finally, for the Scanpaths (Figure 2E, F), a linear temporal gradient ranging from light blue to dark blue was applied, where the radius of each fixation point was mapped to its respective duration. All visualizations are formatted to 1024×512 pixels, which corresponds to two 512×512 unit tiles for the OpenAI api, resulting in a token count of 350 each when using GPT-5.1 [33]. Also, we implemented two text-based prompting approaches for the baselines. *Raw Text* represents the eye-tracking data as a sequence of (x, y) coordinates formatted as comma-separated values. *Feature Text* is a structured symbolic representation which encodes fixations as $F((x, y), t)$ and saccades as $S((x_{\text{start}}, y_{\text{start}}) \rightarrow (x_{\text{end}}, y_{\text{end}}), t)$, where t denotes the duration.

Table 1: Overview of the eye-tracking datasets used in this study.

	GazeBase	*SedentaryActivity*	*DesktopActivity*
Sampling Rate	1000 Hz	30 Hz	30 Hz
Sensor Type	Screen-based	Screen-based	Wearable
Device	EyeLink 1000	Tobii Pro X2-30	Pupil Core
Activity Classes	6	8	6
Participants	14 (Round 9)	24	8
Screen Spec	1680 × 1050 px	24-inch monitor	34-inch monitor
Coord. Unit	dva (deg)	pixel	normalized (0-1)

3.2 Datasets

To assess the generalizability of this approach, we utilize three public eye-tracking datasets:*GazeBase* [34], *SedentaryActivity* [35], and *DesktopActivity* [36]. Each dataset comprises a distinct set of desktop-based activities, collected using either screen-based or egocentric wearable devices. A comprehensive summary of the dataset specifications is provided in Table 1.

GazeBase: This dataset provides longitudinal eye-tracking data collected at 1,000 Hz using EyeLink 1000 [37]. It consists of six activities: *Horizontal Saccade*, *Random Saccade*, *Fixation*, *Reading*, *Video Viewing*, and *Gaze-driven Game*. The original dataset includes two video viewing activities; we utilized only one for distinct class separability. To ensure data stability, we used data from the final session (Round 9) collected from 14 participants. For this dataset only, the *Raw Text* representation was downsampled by 1/10 due to its high sampling rate, considering the model's context limit.

SedentaryActivity: Capturing naturalistic desktop behaviors, this dataset was recorded at 30 Hz using a Tobii Pro X2-30 [38] from 24 participants. It encompasses eight computing tasks: *Read*, *Watch*, *Browse*, *Search*, *Play*, *Interpret*, *Debug*, and *Write*. It contains coding-related tasks (*Interpret*, *Debug*, and *Write*), and each activity is composed of three predefined sub-activities. For this study, we utilized the raw pixel coordinates provided by the tracker.

DesktopActivity: Unlike the previous datasets, this dataset was collected using a wearable egocentric device (Pupil Core [39]) at 30 Hz. Data were gathered from eight participants performing six activities: *Browse*, *Play*, *Read*, *Search*, *Watch*, and *Write*. Due to the absence of specific screen resolution metadata, we utilized the provided normalized coordinates (0 to 1) for our analysis.

3.3 Experimental Design

We first investigated the impact of varying visual prompting strategies for HAR under both zero-shot and one-shot settings. For a one-shot setting, we constructed an example pool by randomly selecting 2, 3, and 1 participants for the datasets, respectively, in proportion to the number of participants in each dataset (14, 24, and 8). The remaining participants served as test data to prevent overlap between the example and test sets. In each experimental run, data from one randomly sampled participant within the example pool were used as the one-shot examples. For evaluation, we generated 30 test cases per activity class. Since the three datasets contain six, eight, and six activity classes, respectively, this resulted in 180, 240, and 180 test cases per dataset for each experimental condition. We used the state-of-the-art MLLM `gpt-5.1-2025-11-13` via the OpenAI API with default inference settings. The `"detail": "high"` parameter was applied only to image inputs.

Furthermore, to evaluate the effect of window size, we conducted additional experiments on the *SedentaryActivity* and *DesktopActivity* datasets, varying the window size from 20 to 100 seconds in 10-second increments. The *GazeBase* dataset was excluded from this analysis as its *Fixations* activity segments are notably short, with an average duration of 14.7 seconds. In these experiments, we focused exclusively on one-shot settings, utilizing four feature-based representations: Timeline, Heatmap, Scanpath, and feature text as a baseline. The selection process for one-shot examples and test cases followed the same protocol as the primary experiment.

Table 2: HAR accuracy and token consumption across experimental conditions (10s window). **Bold** indicates best performance per dataset, and red denotes performance below the textual baseline. Multipliers (× ↑) show the relative increase in token usage of textual prompts compared to visual prompts.

		GazeBase		*SedentaryActivity*		*DesktopActivity*	
		0-shot	1-shot	0-shot	1-shot	0-shot	1-shot
Accuracy							
Timeline	Raw	0.500	**0.811**	0.096	0.283	0.200	0.257
	Feat	0.450	0.772	0.067	0.246	0.200	0.261
Heatmap	Raw	0.578	**0.811**	0.138	**0.311**	0.183	0.300
	Feat	0.644	0.739	0.129	0.304	0.217	0.278
Scanpath	Raw	0.506	0.694	0.183	0.267	0.200	0.300
	Feat	0.544	**0.811**	0.138	0.194	0.272	**0.311**
Text	Raw	0.378	0.539	0.154	0.242	0.189	0.211
	Feat	0.533	0.794	0.167	0.300	0.217	0.244
Number of input tokens							
Visual prompt		1086	3247	1192	4061	1106	3259
Raw text		10748 (9.9× ↑)	70818 (21.8× ↑)	2699 (2.3× ↑)	17678 (4.4× ↑)	3755 (3.4× ↑)	21814 (6.7× ↑)
Feature text		1616 (1.5× ↑)	6828 (2.1× ↑)	1744 (1.46× ↑)	9131 (2.2× ↑)	1617 (1.5× ↑)	6412 (2.0× ↑)

Our prompt design follows the structured framework recommended for IoT foundation models [21]. In the system prompt, we defined the model's role as a domain expert in eye-tracking activity recognition and instructed it to analyze either visual or numerical eye-tracking inputs. And the model was required to output a structured response containing the predicted activity label and a short explanation of the reason for its decision. The user prompt is composed of five segments: `Instruction`, which reiterates the HAR task and expert role; `Activity Descriptions`, which provides informations of each activities and stimuli; `Context`, which includes sensor specifications such as sampling rate and window size; `Examples` (one-shot setting only), which present one representative visualization or text example per activity class; and `Question`, which contains the target instance to be classified. To mitigate ordering bias in language models [40], we randomized the order of activity descriptions and example activities for every query.

4 Results

4.1 Experiment 1: Impact of Visualization Techniques

Table 2 summarizes the HAR accuracy and input token consumption for the various prompting strategies across three datasets. On the *GazeBase* dataset (six classes), the model achieved its highest accuracy of 0.811 under several visual prompting conditions, where clearer performance trends are observed. In contrast, performance on the *SedentaryActivity* (eight classes) and the *DesktopActivity* (six classes) datasets was substantially lower, with the highest accuracies of only 0.311, making performance trends hard to distinguish. These results suggest that a fixed 10-second window is insufficient to capture the activity patterns in these datasets, motivating our subsequent analysis with varying window sizes in Experiment 2.

Compared to textual prompt baselines, visual prompts generally achieved higher performance with raw data. In the one-shot setting, visual prompts using raw data achieved higher accuracy than textual prompts across all datasets and visualization types (Figure 3(A)), and this performance gap is even more apparent for the *GazeBase* dataset, where the Timeline and Heatmap (0.811) significantly outperformed raw text (0.539). In the zero-shot setting, visual prompts continued to outperform text-based prompts, except in a few cases, such as the Timeline and Heatmap on the *SedentaryActivity* and the Heatmap on the *DesktopActivity*. These results imply that transforming raw eye-tracking data into visual representations improves MLLMs' understanding compared to raw textual input.

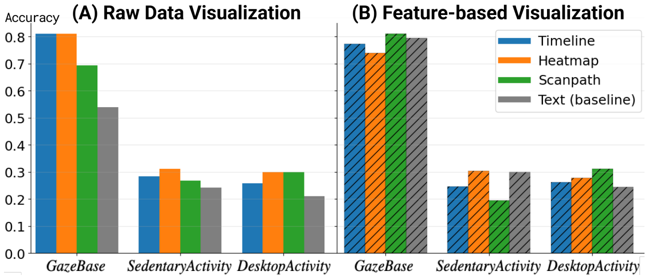

Figure 3: One-shot setting accuracy comparison between visualization techniques across three datasets.

However, this advantage does not consistently hold when the visual prompts are compared against feature-based textual representations. As shown in Figure 3(B), the feature text prompt exhibited performance comparable to visual prompts across three datasets. In the *SedentaryActivity*, for instance, the feature text prompt outperformed most visual prompting conditions both in zero-shot and one-shot settings. Similarly, in the *GazeBase* one-shot setting, only the Scanpath surpassed the feature text accuracy. These results indicate that explicit gaze features are effective even in text form, compared to raw textual sequences. From another perspective, results on the *GazeBase* show that raw Timeline and Heatmap visualizations achieve accuracy comparable to the feature text prompt. This suggests that in some cases, appropriate visual abstractions can bypass the need for explicit feature extraction.

When comparing raw data and feature-based representations within the same visualization type, their accuracies exhibited distinct differences. For the Timelines, using raw data generally yielded higher accuracy than feature-based visualizations, except for the *DesktopActivity* one-shot setting. In contrast, the Heatmap showed negligible performance differences between data types. The Scanpaths exhibited the most noticeable variance between data types. For example, in the *GazeBase* one-shot setting, the feature-based Scanpath (0.811) outperformed the raw Scanpath (0.694), whereas the opposite trend was observed in the *SedentaryActivity* dataset, where the raw Scanpath (0.267) outperformed the feature-based Scanpath (0.194). We attribute this pattern to the dataset-specific activity characteristics, which influence the effectiveness of explicit fixation and saccade encoding.

In terms of token consumption, visual prompting offers a substantial advantage. As detailed in Table 2, visual prompts consumed fewer tokens than textual prompts. The raw text baseline required between 2.3× and 21.8× more tokens than visual prompts to represent the same 10-second window. Even the more condensed feature text consumed 1.46× to 2.2× more tokens. These findings highlight visual prompting as a cost-effective approach for processing high-frequency eye-tracking data with MLLMs.

4.2 Experiment 2: Impact of Window Size

Figure 4 presents the classification accuracy and token consumption results for the *SedentaryActivity* and *DesktopActivity* datasets across temporal window sizes ranging from 10 to 100 seconds. In the *SedentaryActivity* dataset, accuracy generally improved as the window size increased. The Heatmap achieved the highest accuracy across nearly all window sizes except for the 20-second condition. Performance for the Heatmap showed signs of saturation beyond 60 seconds, maintaining a stable range between 0.521 and 0.529. The performance ranking remained stable (Heatmap, Scanpath, Timeline, and the textual baseline) for window sizes of 40 seconds or longer, except at 70 seconds, where Timeline and Scanpath switched orders. Unlike the Heatmap, the Scanpath visualization exhibited a steady upward trend without reaching a clear saturation point, peaking at 0.4875 at 100-second. Among all visualization types, only the Timeline exhibited a peak in performance and reached an accuracy of 0.450 at 70 seconds before declining to 0.392 at 100 seconds. Although the textual baseline also showed improvement with larger windows, its maximum accuracy (0.429 at 90 seconds) remained lower than the visual prompts. Given the eight activity classes in this dataset, the highest accuracy of 0.529 indicates that the model can capture relevant behavioral patterns when provided with sufficient temporal context. Taken together, the results suggest that extended temporal windows combined with appropriate visual representations facilitate more effective utilization of gaze dynamics in the *SedentaryActivity* dataset.

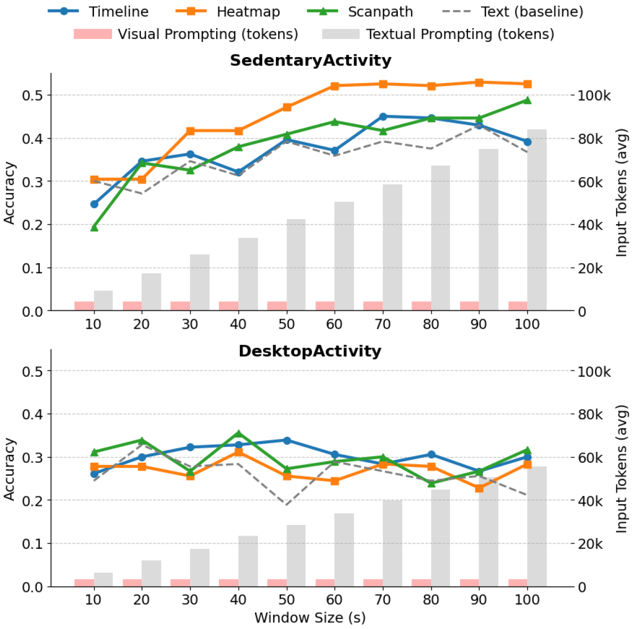

Figure 4: Comparison of HAR accuracy and token consumption across varying temporal window sizes (10-second to 100-second) for the *SedentaryActivity* and *DesktopActivity* datasets. The line plots illustrate the accuracy trends for each visual prompting strategy (Timeline, Heatmap, Scanpath, and the textual baseline). The grouped bar charts represent the corresponding input tokens for each condition.

In contrast, the *DesktopActivity* dataset did not exhibit a clear accuracy pattern related to window size. The accuracy remained around 0.3 under all conditions. The highest performances were recorded at different intervals: 0.339 at a 50-second window for the Timeline, 0.356 at a 40-second window for the Scanpath, and 0.311 at a 40-second window for the Heatmap. In particular, the Heatmap showed relatively poor performance compared to other visual prompts in this dataset, which contradicts the results seen in the *SedentaryActivity*. Furthermore, no distinct performance gap was observed between visual and textual prompting strategies in this specific dataset. These results suggest that neither window size nor prompt modality strongly influences accuracy in this dataset.

Regarding token usage, the visual prompts offer a clear advantage over the textual prompts. Token consumption for textual prompts scales linearly with temporal duration, whereas visual prompts maintain a constant token count regardless of window size. In the *SedentaryActivity* dataset, visual prompts consistently utilized 4,061 tokens. At the 100-second window, the textual prompt required 83,901 tokens, corresponding to a 20.7× increase over the visual prompt. Similarly, for the *DesktopActivity* dataset, visual prompts used a fixed 3,259 tokens, whereas the textual baseline reached 55,481 tokens at 100 seconds, representing a 17.0× increase. These results demonstrate that visual prompting provides a scalable approach for analyzing long-duration sensor data without the proportional increase in computational cost associated with textual representations.

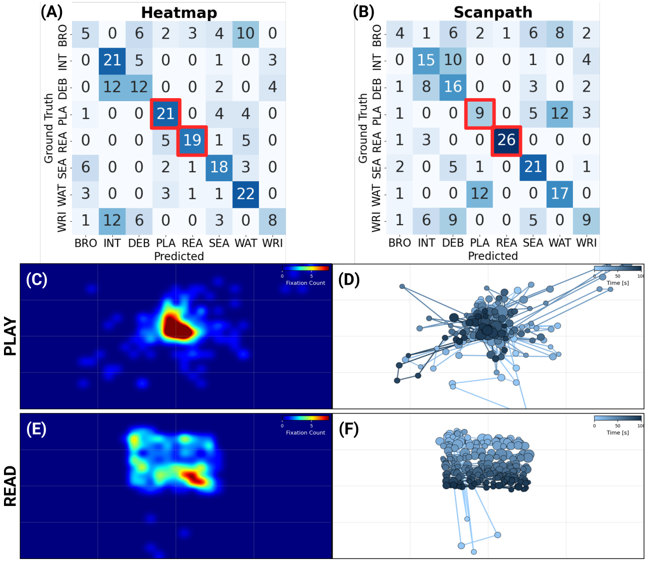

Figure 5: Confusion matrices for (A) Heatmap and (B) Scanpath visual prompting, alongside representative examples for PLAY (C, D) and READ (E, F) from the *SedentaryActivity* 100-second window experiment. Activity names in the matrices are abbreviated to the first three characters. Titles, axis titles and ticks for the examples (C–F) are omitted to enhance visual clarity.

5 Discussions

A primary advantage of visual prompting lies in its scalability and token efficiency. In the *GazeBase* dataset, which consists of experimental tasks such as *Horizontal Saccade* and *Fixation*, a 10-second window was sufficient for the model to distinguish between classes. In contrast, the *SedentaryActivity* dataset comprises more complex and naturalistic activities (e.g., *Search*, *Debug*), where applying larger windows to include more behavioral information proved more effective. This observation aligns with the findings of Lan et al. [36], who reported that recognition performance increased as the sensing window size grew from 10 to 30 seconds in their graph-based gaze representation learning framework. While providing extended temporal context is beneficial for these complex activities, textual prompting suffers from the "lost in the middle" problem [8] and reduced computational efficiency. Visual prompting can overcome these limitations by maintaining a constant token cost within a fixed canvas, regardless of the data length. In particular, the Timeline representations can offer broader extensibility, as their temporal structure can be readily adapted to other time-series sensor modalities (e.g., IMU or physiological signals).

However, this efficiency involves a trade-off with information density. As observed in the Timeline plots for the *SedentaryActivity* dataset, accuracy may decline when the temporal window exceeds a certain threshold due to visual overcrowding and subsequent information loss. This suggests that maintaining an optimal information density is a critical factor in designing effective visual prompts for high-frequency sensor data. Finally, the visual prompting approach did not yield significant performance gains in the *DesktopActivity* dataset, which may be caused by its unique characteristics such as the use of a egocentric sensor or potential overfitting to the provided examples, which necessitates further investigation.

Furthermore, no single visualization technique or data type (raw vs. feature data) consistently demonstrated superior performance across all conditions. The effectiveness of a specific visualization appears to be highly dependent on the behavioral characteristics of the target activity. In the 100-second window experiment for the *SedentaryActivity* dataset (Figure 5), the *Play* activity was more effectively classified using the Heatmap (Figure 5(C)) than the Scanpath (Figure 5(D)) (21 vs. 9). In contrast, the *Read* activity showed higher accuracy with the Scanpath (Figure 5(F)) compared to the Heatmap (Figure 5(E)) (26 vs. 19). This contrast indicates that different activities emphasize different aspects of gaze behavior. For *Play*, the spatial distribution and density of attention are more informative, while for *Read*, the temporal ordering of gaze movements plays a more central role. These results highlight the importance of choosing visualizations for MLLMs that align with the task-specific behavioral structure. Our findings can contribute to the design of an autonomous agent capable of selecting the most representative eye-tracking visualization for a given context. Additionally, combining multiple visualization techniques to integrate their strengths may offer improvements and warrants further investigation.

While the HAR accuracy of visual prompting currently remains lower than that of conventional approaches [36], it offers a training-free alternative that leverages the pre-existing world knowledge and visual reasoning capabilities of MLLMs [21]. By utilizing visual encoders to interpret gaze patterns, this method provides a novel framework for utilizing data visualization as a bridge between human context and foundation models in the IoT ecosystem. Beyond HAR, this framework may support higher-level inference about user attention or task engagement, which are central to pervasive systems in ubiquitous computing scenarios. Rather than limiting the model to predicting predefined activity labels, visual prompts could support more flexible reasoning, such as inferring whether a user is focused, distracted, or searching for a component. In our study, we evaluated visual prompting on three datasets covering a diverse range of activities; however, future investigation is needed to assess its applicability in more open-ended real-world IoT scenarios. Taken together, our findings suggest that visual prompting may serve as a flexible intermediate representation for context-aware reasoning without task-specific retraining, while its generalizability requires further empirical validation.

6 Conclusion

In this study, we investigated the efficacy of visual prompting for eye-tracking-based HAR using MLLMs. We designed and evaluated three visualizations across three public datasets and multiple temporal window sizes. Our findings provide a valuable insight that visual prompting serves as a token-efficient and scalable alternative to textual prompting for MLLMs, especially for long duration eye-tracking data. We discussed the importance of task-specific visualization selection and the applicability of sensor data visualization in foundation model contexts. This study has several limitations, including reliance on closed-source models, the lack of direct comparisons with conventional HAR methods, and its focus being limited to HAR evaluation within IoT tasks. Finally, future work will explore real-world application scenarios that leverage visual prompting as a interface for MLLM-based IoT systems.

Supplemental Materials

Supplemental materials[2] provide all visual prompts used in the experiments to support further analysis and reproducibility.

References

[1] H. Xu, L. Han, Q. Yang, M. Li, and M. Srivastava, "Penetrative ai: Making llms comprehend the physical world," in *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*, pp. 1–7, 2024. 1

[2] S. Ji, X. Zheng, and C. Wu, "Hargpt: Are llms zero-shot human activity recognizers?," in *2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys)*, pp. 38–43, IEEE, 2024. 1, 2

[3] J. Li, X. Li, J. Steinberg, A. Choube, B. Yao, X. Xu, D. Wang, E. Mynatt, and V. Mishra, "Vital insight: Assisting experts' context-driven

[2] Available at https://eyetrackingvisualprompts.github.io.

sensemaking of multi-modal personal tracking data using visualization and human-in-the-loop llm," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 3, pp. 1–37, 2025. 1

[4] X. Liu, D. McDuff, G. Kovacs, I. Galatzer-Levy, J. Sunshine, J. Zhan, M.-Z. Poh, S. Liao, P. Di Achille, and S. Patel, "Large language models are few-shot health learners," *arXiv preprint arXiv:2305.15525*, 2023. 1

[5] B. Yang, S. Jiang, L. Xu, K. Liu, H. Li, G. Xing, H. Chen, X. Jiang, and Z. Yan, "Drhouse: An llm-empowered diagnostic reasoning system through harnessing outcomes from sensor data and expert knowledge," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 4, pp. 1–29, 2024. 1

[6] Q. Wei, J. Huang, Y. Gao, and W. Dong, "One model to fit them all: Universal imu-based human activity recognition with llm-assisted cross-dataset representation," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 3, pp. 1–22, 2025. 1, 2

[7] D. Spathis and F. Kawsar, "The first step is the hardest: Pitfalls of representing and tokenizing temporal data for large language models," *Journal of the American Medical Informatics Association*, vol. 31, no. 9, pp. 2151–2158, 2024. 1

[8] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024. 1, 5

[9] T. Zhou, P. Niu, L. Sun, R. Jin, *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43322–43355, 2023. 1

[10] 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*, pp. 220–233, 2021. 1

[11] S. Zhao, S. Zhou, R. Blanchard, Y. Qiu, W. Wang, and S. Scherer, "Tartan imu: A light foundation model for inertial positioning in robotics," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22520–22529, 2025. 1

[12] H. Yoon, B. Tolera, T. Gong, K. Lee, and S.-J. Lee, "By my eyes: Grounding multimodal large language models with sensor data via visual prompting," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2219–2241, 2024. 1, 2

[13] J. P. Hansen, H. Lund, F. Biermann, E. Møllenbach, S. Sztuk, and J. S. Agustin, "Wrist-worn pervasive gaze interaction," in *Proceedings of the ninth biennial ACM symposium on eye tracking research & applications*, pp. 57–64, 2016. 1

[14] M. Barz, S. Kapp, J. Kuhn, and D. Sonntag, "Automatic recognition and augmentation of attended objects in real-time using eye tracking and a head-mounted display," in *ACM Symposium on Eye Tracking Research and Applications*, pp. 1–4, 2021. 1

[15] D. Noton and L. Stark, "Scanpaths in saccadic eye movements while viewing and recognizing patterns," *Vision research*, vol. 11, no. 9, pp. 929–IN8, 1971. 1, 2

[16] J. F. Mackworth and N. H. Mackworth, "Eye fixations recorded on changing visual scenes by the television eye-marker," *Journal of the Optical Society of America*, vol. 48, no. 7, pp. 439–445, 1958. 1, 2

[17] Z. Li, S. Deldari, L. Chen, H. Xue, and F. D. Salim, "Sensorllm: Aligning large language models with motion sensors for human activity recognition," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 354–379, 2025. 2

[18] Z. Hong, Y. Song, Z. Li, A. Yu, S. Zhong, Y. Ding, T. He, and D. Zhang, "Llm4har: Generalizable on-device human activity recognition with pretrained llms," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4511–4521, 2025. 2

[19] L. Xu, K. Hou, and X. Jiang, "Exploring the capabilities of llms for imu-based fine-grained human activity understanding," in *Proceedings of the 2nd International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things*, pp. 13–18, 2025. 2

[20] H. Yan, H. Tan, Y. Ding, P. Zhou, V. Namboodiri, and Y. Yang, "Large language model-guided semantic alignment for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 4, pp. 1–25, 2025. 2

[21] H. Wei, D. Y. Lee, S. Rohal, Z. Hu, R. Rossi, S. Fang, and S. Pan, "A survey of foundation models for iot: taxonomy and criteria-based analysis: H. wei et al.," *CCF Transactions on Pervasive Computing and Interaction*, pp. 1–29, 2025. 2, 3, 5

[22] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "Visualization of eye tracking data: A taxonomy and survey," in *Computer graphics forum*, vol. 36, pp. 260–284, Wiley Online Library, 2017. 2

[23] K.-J. Räihä, A. Aula, P. Majaranta, H. Rantala, and K. Koivunen, "Static visualization of temporal eye-tracking data," in *IFIP Conference on Human-Computer Interaction*, pp. 946–949, Springer, 2005. 2

[24] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye tracking: A comprehensive guide to methods and measures*. oup Oxford, 2011. 2

[25] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 71–78, 2000. 2

[26] J. H. Goldberg and J. I. Helfman, "Visual scanpath representation," in *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pp. 203–210, 2010. 2

[27] T. Grindinger, A. T. Duchowski, and M. Sawyer, "Group-wise similarity and classification of aggregate scanpaths," in *Proceedings of the 2010 symposium on eye-tracking research & applications*, pp. 101–104, 2010. 2

[28] A. Bojko, "Informative or misleading? heatmaps deconstructed," in *International conference on human-computer interaction*, pp. 30–39, Springer, 2009. 2

[29] L. F. Scinto, R. Pillalamarri, and R. Karsh, "Cognitive strategies for visual search," *Acta psychologica*, vol. 62, no. 3, pp. 263–292, 1986. 2

[30] C. Lankford, "Gazetracker: software designed to facilitate eye movement analysis," in *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 51–55, 2000. 2

[31] K. Kurzhals and D. Weiskopf, "Space-time visual analytics of eye-tracking data for dynamic stimuli," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2129–2138, 2013. 2

[32] J. Y. Choi, S. G. Kim, J. Jeong, R. A. Rossi, J. Kil, and T. Y. Lee, "Gaze2prompt: Turning eye-tracking data into visual prompts for multimodal llms," in *Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 110–114, 2025. 2

[33] OpenAI, "Api pricing," 2025. https://openai.com/api/pricing/. Accessed: 2026-01-03. 2

[34] H. Griffith, D. Lohr, E. Abdulin, and O. Komogortsev, "Gazebase, a large-scale, multi-stimulus, longitudinal eye movement dataset," *Scientific Data*, vol. 8, no. 1, p. 184, 2021. 3

[35] N. Srivastava, J. Newn, and E. Velloso, "Combining low and mid-level gaze features for desktop activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–27, 2018. 3

[36] G. Lan, B. Heit, T. Scargill, and M. Gorlatova, "Gazegraph: Graph-based few-shot cognitive context sensing from human visual behavior," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pp. 422–435, 2020. 3, 5

[37] SR Research Ltd., "Eyelink 1000 user's manual," 2010. Version 1.5.2. 3

[38] Tobii, "Tobii pro x2-30 eye tracker." https://www.tobii.com/products/discontinued/tobii-pro-x2-30/. Accessed: 2026-01-03. 3

[39] Pupil Labs, "Pupil core." https://pupil-labs.com/. Accessed: 2026-01-03. 3

[40] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in neural information processing systems*, vol. 36, pp. 46595–46623, 2023. 3