SensorLLM: Aligning Large Language Models with Motion Sensors for Human Activity Recognition

Anonymous ACL submission

Abstract

We introduce SensorLLM, a two-stage framework that enables Large Language Models (LLMs) to perform human activity recognition (HAR) from sensor data. Despite their strong reasoning and generalization capabilities, LLMs remain underutilized for motion sensor data due to the lack of semantic context in time-series, computational constraints, and challenges in processing numerical inputs. SensorLLM addresses these limitations through a Sensor-Language Alignment stage, where we introduce special tokens for each sensor channel and automatically generate textual trend descriptions. This alignment enables LLMs to capture numerical variations, channel-specific features, and data of varying durations—without requiring human annotations. In the subsequent Task-Aware Tuning stage, we refine the model for HAR classification, achieving performance that matches or surpasses state-of-the-art methods. Our results demonstrate that SensorLLM evolves into an effective sensor learner, reasoner, and classifier through Sensor-Language Alignment, generalizing across diverse HAR datasets. We believe this work establishes a foundation for future research on time-series and text alignment, paving the way for foundation models in sensor data analysis. Our codes are available at https://anonymous.4open.science/r/sensorllm_code-E0FC.

1 Introduction

Human Activity Recognition (HAR) is a time-series classification task that maps sensor signals, such as accelerometer and gyroscope data, to human activities. Traditional models like LSTM (Guan and Plötz, 2017; Hammerla et al., 2016) and DeepConvLSTM (Ordóñez and Roggen, 2016) learn high-level features but are task-specific and struggle to generalize across different sensor configurations and activity sets. In contrast, Large Language Models (LLMs) (Han et al., 2021) have shown remarkable success in integrating diverse data types (Wu et al., 2023b; Yin et al., 2023), including text and images.

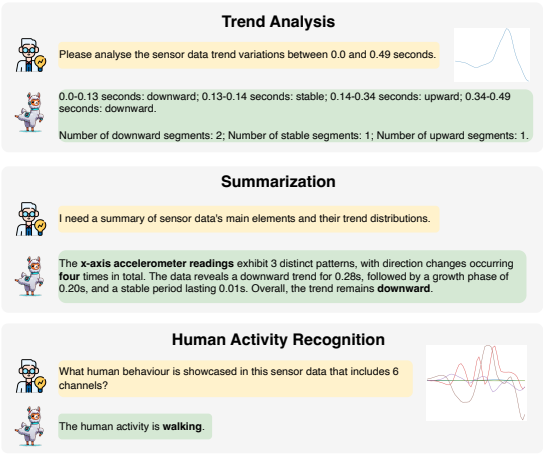

Figure 1: SensorLLM can analyze and summarize trends in captured sensor data, facilitating human activity recognition tasks.

Enabling LLMs to process sensor data (Jin et al., 2023) requires either (1) pretraining or fine-tuning on time-series data (Zhou et al., 2023a), which demands substantial computational resources and is hindered by limited and imbalanced labeled data, or (2) leveraging zero-shot and few-shot prompting by converting sensor data into text (Kim et al., 2024; Ji et al., 2024). The latter approach avoids retraining but introduces key challenges: (i) Numerical encoding issues—LLM tokenizers, designed for text, struggle with numerical values, treating consecutive numbers as independent tokens (Nate Gruver and Wilson, 2023) and failing to preserve temporal dependencies (Spathis and Kawsar, 2024). (ii) Sequence length constraints—sensor data often exceeds LLMs' maximum context length, leading to truncation, information loss, and increased computational costs. (iii) Multi-channel complexity—LLMs process univariate inputs, making it difficult to encode multi-sensor data in a way that retains inter-channel dependencies. (iv) Prompt en-

gineering challenges—designing effective prompts that enable LLMs to interpret numerical sensor readings, detect trends and classify activities remains a challenge (Liu et al., 2023b).

To address these challenges, we propose SensorLLM, a framework that aligns sensor data with natural language, allowing LLMs to analyze sensor data through text-based interactions (see Figure 1) without modifying the LLM itself. A major challenge is the annotation bottleneck, further compounded by the complexity and heterogeneity of sensor data. Unlike image-text pairs, sensor data comprises multi-channel numerical signals with diverse characteristics, making direct interpretation and alignment difficult.

Existing methods (Jin et al., 2024a; Sun et al., 2024a) have explored condensed text prototypes for alignment, but these approaches often lack interpretability and require extensive tuning to select suitable prototypes. In contrast, we propose an automatic text generation approach that aligns with human intuition by deriving descriptive trend-based text directly from time-series data using statistical analyses and predefined templates. This method is precise, scalable, and interpretable, eliminating the need for manual annotations while preserving essential sensor characteristics. SensorLLM follows a two-stage framework:

Sensor-Language Alignment Stage. We automatically generate question-answer pairs to align sensor data with text while preserving temporal features using a pretrained encoder. The resulting embeddings are mapped into a space interpretable by the LLM, mitigating issues associated with text-specific tokenization. Additionally, we introduce *special tokens* for sensor channels, enabling LLMs to effectively capture multi-channel dependencies.

Task-Aware Tuning Stage. The aligned embeddings are utilized for HAR, leveraging the LLM's reasoning capabilities while keeping its parameters frozen. This design extends LLMs beyond their original training, addressing concerns raised by Tan et al. (2024) regarding their applicability to time-series data. To our knowledge, this is the first approach to integrate sensor data into LLMs for sensor data analysis and HAR tasks.

To our knowledge, this is the first approach to integrate sensor data into LLMs for sensor data analysis and HAR tasks. The key contributions of this work are as follows:

- We propose a fully automated, human-intuitive approach for aligning time-series data with descriptive text, eliminating the need for manual annotations. Using text similarity metrics, human evaluations, and LLM-based assessments, we demonstrate that SensorLLM effectively captures temporal patterns and channel-specific features, enabling robust multimodal understanding.

- SensorLLM achieves competitive results across five HAR datasets, matching or surpassing state-of-the-art models. Experiments further validate that *modality alignment* and *task-specific prompts* significantly enhance the LLM's ability to interpret and classify sensor data.

- We show that SensorLLM maintains strong performance in the Task-Aware Tuning Stage, even when applied to datasets distinct from those used during alignment, highlighting its robustness and generalizability in HAR tasks.

2 Related Work

In this section, we discuss recent developments in leveraging LLMs for time-series data, specifically focusing on two categories: (1) LLMs for time series as text and (2) Multimodal Large Language Models (MLLMs) for sensor data. A broader overview of other related works, including deep learning approaches to HAR and additional LLM-based forecasting methods, is provided in Appendix A.1.

LLMs for Time Series as Text. While LLMs excel in processing natural language, applying them directly to time-series data poses unique challenges (Spathis and Kawsar, 2024). Certain methods address this by treating time-series signals as raw text, using the same tokenization as natural language. Notable examples include PromptCast (Xue and Salim, 2023), which transforms numeric inputs into textual prompts for zero-shot forecasting, and LLMTime (Gruver et al., 2024), which encodes time-series as numerical strings for GPT-like models. However, due to the lack of specialized tokenizers for numeric sequences, LLMs may fail to capture crucial temporal dependencies and repetitive patterns (Spathis and Kawsar, 2024). To mitigate these issues, several works employ time-series encoders before mapping the resulting embeddings to language model spaces (Liu et al., 2024a; Zhou

et al., 2023c; Xia et al., 2024), thus aligning sensor embeddings with textual embeddings in a contrastive or supervised manner.

MLLMs for Sensor Data. Extending LLMs to non-textual domains has gained traction, particularly through MLLMs that accept inputs beyond text, such as images or speech. For sensor data, the challenge lies in representing continuous signals effectively. Yoon et al. (2024) propose to ground MLLMs with sensor data via visual prompting. Sensor signals are first visualized as images, guiding the MLLM to analyze the visualized sensor traces alongside task descriptions, which also lower token costs compared to raw-text baselines. Similarly, Moon et al. (2023) introduce IMU2CLIP, which aligns inertial measurement unit streams with text and video in a joint representation space. This approach enables wearable AI applications like motion-based media search and LM-based multimodal reasoning, showcasing how sensor data can be integrated into broader multimodal frameworks.

3 Methods

In this work, we propose SensorLLM, a framework that aligns sensor data with descriptive text through automatically generated question–answer pairs. Our goal is to develop a multimodal model with reasoning capabilities for analyzing wearable sensor data. As shown in Figure 2, SensorLLM comprises three core components: (1) a pretrained LLM, (2) a pretrained time-series (TS) embedder, and (3) an alignment module MLP, all operating in a two-stage framework.

In the Sensor-Language Alignment Stage, a generative model aligns sensor readings with text based on user instructions (Liu et al., 2023a) and questions. In the Task-Aware Tuning Stage, a classifier is added on top of the LLM to perform HAR. Crucially, only the MLP in both stages and the classifier in the Task-Aware Tuning Stage are trainable, while the backbone LLM and TS embedder remain frozen. This design yields a highly efficient and lightweight training process, requiring only 5.67% (535.9M) of the parameters to be trainable in the alignment stage and 0.12% (10.5M) in the tuning stage.

3.1 Sensor-Text Data Generation

Aligning time-series data with text for LLM-based tasks is challenging due to the lack of rich semantic labels beyond class annotations, making manual annotation impractical (Deldari et al., 2024; Haresamudram et al., 2024). While prior works rely on predefined text prototypes (Sun et al., 2024b; Jin et al., 2024a), we aim for a more human-intuitive representation of sensor data.

We argue that time-series data inherently contains semantic patterns that can be expressed through descriptive text, from simple numerical trends to statistical insights. To achieve this, we automatically generate descriptive text by analyzing observed trends and fluctuations in the data. Using predefined templates, we construct diverse question-answer (QA) pairs that capture trend changes while ensuring accuracy and scalability. These templates (Appendix A.2) are randomly combined to enhance diversity. For example:

(1) The time-series data represents readings taken from a <S> sensor between <t_s> and <t_e> seconds.

(2) To sum up, the data exhibited a <T> trend for a cumulative period of <t_t> seconds.

where `T` and `S` denote specific trends and sensor types, and `t` corresponds to numerical values.

3.2 Sensor-Language Alignment

As shown in Figure 2 (a), the Sensor-Language Alignment stage employs a generative model to create multimodal sentences that combine single-channel sensor readings with textual descriptions. The sensor data is represented as a matrix $\mathbf{X} \in \mathbb{R}^{C \times T}$, where C is the number of sensor channels and T is the sequence length. Each channel's data, denoted as $\mathbf{X}^c$ for channel c, is processed independently to preserve its unique characteristics. The data is then divided into non-overlapping segments, $\mathbf{X}_S^c$, where S is the total number of segments. Each segment x_s is assigned a random length l within a predefined range, allowing the model to learn from varying temporal patterns and trend variations. This segmentation strategy ensures that both long-term trends and short-term fluctuations are effectively captured in the generated multimodal sentences.

We use Chronos (Ansari et al., 2024) as the TS embedder, generating segment embeddings $\hat{x}_s \in \mathbb{R}^{(l+1) \times d_{ts}}$, where d_{ts} is the feature dimension, and $(l+1)$ accounts for the `[EOS]` token appended during Chronos tokenization. Prior to inputting sensor segments into Chronos (Appendix A.3), we apply

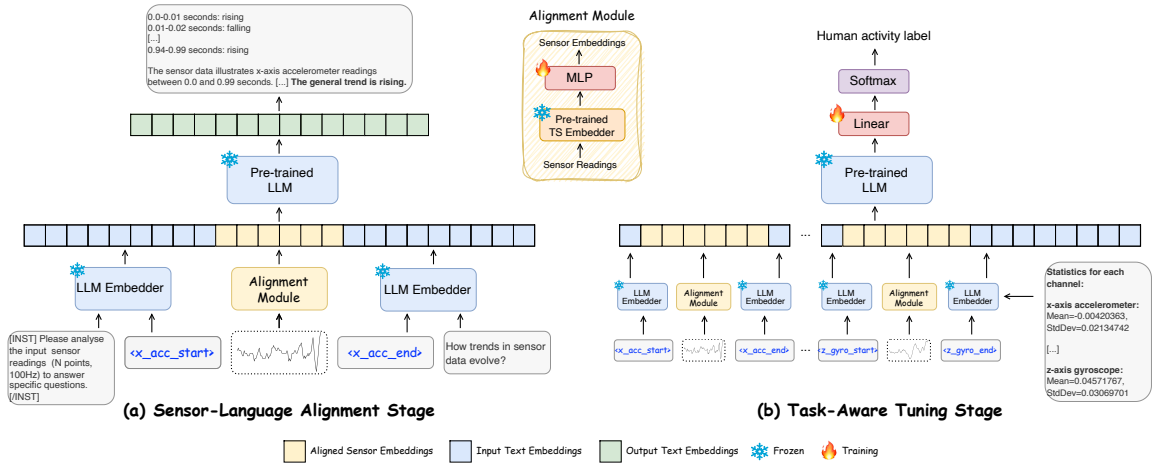

Figure 2: Our proposed SensorLLM framework: **(a) Sensor-Language Alignment Stage**, where a generative model aligns sensor readings with automatically generated text; **(b) Task-Aware Tuning Stage**, where a classification model leverages the aligned modalities to perform HAR.

instance normalization $\tilde{x}_s = \frac{x_s - \text{mean}(x_s)}{\text{std}(x_s)}$ to standardize the data. Llama3-8B (Touvron et al., 2023) serves as our LLM backbone.

Alignment Module. To transform TS embeddings $\hat{x}_s$ into text-aligned embeddings $\hat{a}_s \in \mathbb{R}^{(l+1)\times D}$ for downstream tasks, we introduce an alignment projection module. This module, implemented as a multi-layer perceptron (MLP), first maps sensor embeddings to an intermediate space of dimension d_m and then projects them to the target dimension D. Formally,

$$\hat{a}_s = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \hat{x}_s + \mathbf{b}_1) + \mathbf{b}_2, \qquad (1)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_m \times d_{ts}}$ and $\mathbf{W}_2 \in \mathbb{R}^{D \times d_m}$ are learnable weights, $\mathbf{b}_1$ and $\mathbf{b}_2$ are biases, and σ is the GELU activation function (Hendrycks and Gimpel, 2016). This projection ensures that the transformed embeddings $\hat{a}_s$ are semantically aligned with the text embedding space, making them suitable for tasks such as text generation and classification.

Input Embedding. To integrate sensor data into the LLM, we introduce two special tokens per sensor channel (e.g., <x_acc_start> and <x_acc_end> for the x-axis accelerometer), extending the LLM's embedding matrix from $\mathbf{E} \in \mathbb{R}^{V \times D}$ to $\mathbf{E} \in \mathbb{R}^{V' \times D}$, where $V' = V + 2c$, with V as the vocabulary size and c as the number of channels. These special token embeddings are concatenated with the aligned sensor embeddings. The final combined sensor representation $\hat{o}_s \in \mathbb{R}^{(l+3)\times D}$ is then concatenated with instruction and question embeddings to form the full input sequence $\hat{z} \in \mathbb{R}^{k \times D}$, where k is the total number of tokens.

Loss Function. SensorLLM processes an input sequence $\mathbf{Z}_s = \{z_s^i\}_{i=1}^K$ consisting of sensor and text embeddings and generates an output sequence $\mathbf{Z}_t = \{z_t^i\}_{i=1}^N$, where $z_s^i, z_t^i \in V'$, and K and N represent the number of input and output tokens, respectively. The model is trained using a causal language modeling objective, predicting the next token based on previous ones. The optimization minimizes the negative log-likelihood:

$$\mathcal{L}_{gen} = -\sum_{i=0}^{N-1} \log P(z_t^i | Z_t^{<i}, z_s). \qquad (2)$$

Loss is computed only on generated tokens, ensuring SensorLLM effectively integrates sensor and text embeddings to produce coherent, contextually appropriate responses.

3.3 Task-Aware Tuning

As shown in Figure 2 (b), the Task-Aware Tuning stage refines the multimodal sensor-text embeddings for HAR. This stage integrates multi-channel sensor readings with activity labels, aligning temporal patterns with human activities. The input sensor data $\mathbf{X}$ is segmented into overlapping windows of size L with a 50% overlap (Li et al., 2018), forming segments $\mathbf{X}_S \in \mathbb{R}^{S \times C \times L}$, where S is the number of segments and C is the number of channels. The pretrained alignment module from the first stage maps sensor data to activity labels, preserving inter-channel dependencies while learning activity-related patterns.

Input Embedding. For each sensor channel c, we retrieve its aligned sensor embeddings $\hat{o}_s^c$. These embeddings are then concatenated across all channels, along with their corresponding statistical features (mean and variance), to form the final input embedding:

$$\hat{z} = \hat{o}_s^1 \oplus \hat{o}_s^2 \oplus \cdots \oplus \hat{o}_s^C \oplus \hat{z}_{\text{stat}}, \tag{3}$$

where $\hat{z}_{\text{stat}}$ represents the statistical information, and C is the number of channels. This ensures the model integrates both temporal and statistical characteristics for HAR.

Loss Function. The input token sequence is processed by the LLM, yielding a latent representation $\mathbf{H} \in \mathbb{R}^{K \times D}$, where K is the number of tokens and D is the embedding dimension. Due to causal masking, we extract the final hidden state, $\mathbf{h} = \mathbf{H}_K$, which encodes all preceding token information. This pooled vector is passed through a fully connected layer to produce a prediction vector of size M, where M is the number of activity classes. The final class probabilities $\hat{y}_i$ are obtained via the softmax function, and the model is optimized using cross-entropy loss:

$$\mathcal{L}_{cls} = -\sum_{i=0}^{M-1} y_i \log \hat{y}_i, \tag{4}$$

where y_i is the ground truth label.

4 Experiments

In this section, we evaluate SensorLLM in enabling LLMs to interpret, reason about, and classify sensor data for HAR tasks. All experiments are conducted on NVIDIA A100-80G GPUs. To assess the LLM's ability to learn and generalize from raw sensor inputs, we ensure that the same training and testing subjects are used in both the Sensor-Language Alignment and Task-Aware Tuning stages. This guarantees that test data in the second stage remains unseen during alignment, ensuring a fair evaluation of generalization. We select Chronos as the TS embedder because it has not been pre-trained on motion sensor data, making it an ideal candidate for evaluating our approach's robustness in learning directly from unprocessed sensor signals.

4.1 Datasets

To evaluate the effectiveness and generalizability of SensorLLM, we conduct experiments on five publicly available HAR datasets: USC-HAD (Zhang and Sawchuk, 2012), UCI-HAR (Anguita et al., 2013), PAMAP2 (Reiss and Stricker, 2012), MHealth (Baños et al., 2014), and CAPTURE-24 (Chan et al., 2024). These datasets differ in sensor placement, sampling rates, channel configurations, and activity types, covering both controlled laboratory conditions and free-living environments. Additionally, they vary in scale, with USC-HAD, UCI-HAR, PAMAP2, and MHealth collected from a limited number of subjects, whereas CAPTURE-24 features a large-scale dataset with sensor recordings from 151 participants in real-world settings. Full dataset details, including subject count, sensor configurations, data splits, activity classes, preprocessing steps, and windowing strategies, are provided in Appendix A.6.

4.2 Sensor Data Understanding

Setup. We train SensorLLM on all five datasets using the same following parameters in the Sensor-Language Alignment Stage: a learning rate of 2e-3, 8 epochs, batch size of 4, gradient accumulation steps of 8, and a maximum sequence length of 8192 for CAPTURE-24 and 4096 for the other datasets.

Evaluation Metrics. We assess SensorLLM's ability to generate trend descriptions from sensor data, comparing it with the advanced GPT-4o [1] to evaluate Sensor-Language Alignment. GPT-4o generates responses based on a predefined prompt (Appendix A.4), following our text template. We employ three evaluation methods:

- **NLP Metrics.** We measure surface-level similarity and n-gram overlap using BLEU-1 (Papineni et al., 2002), ROUGE-1, ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). We also use SBERT (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021) to evaluate semantic similarity between model outputs and ground truth.

- **GPT-4o Evaluation.** GPT-4o rates the generated trend descriptions on a scale of 1 to 5 (with 5 being the highest) by comparing each output to ground truth and providing explanatory feedback. As an advanced LLM, its evaluation ensures a semantic assessment of trend comprehension.

- **Human Evaluation.** Five time-series experts (PhD students, postdocs, and academics)

[1] gpt-4o-2024-08-06(OpenAI, 2024)

Metric	USC-HAD		UCI-HAR		PAMAP2		MHealth		CAPTURE-24	
	GPT-4o	Ours	GPT-4o	Ours	GPT-4o	Ours	GPT-4o	Ours	GPT-4o	Ours
BLEU-1	41.43	**57.68**	37.97	**56.78**	46.35	**60.20**	49.97	**61.38**	46.58	**57.10**
ROUGE-1	54.92	**68.32**	51.24	**67.63**	58.08	**69.92**	61.11	**71.20**	58.21	**68.11**
ROUGE-L	49.00	**64.17**	44.88	**63.05**	50.30	**66.25**	51.99	**67.83**	48.88	**60.90**
METEOR	30.51	**45.95**	26.93	**45.81**	37.17	**52.21**	38.50	**51.73**	31.16	**40.51**
SBERT	77.22	**86.09**	76.05	**85.01**	82.71	**87.31**	83.15	**86.66**	83.11	**84.83**
SimCSE	86.96	**93.09**	90.23	**92.51**	89.64	**93.82**	92.10	**93.38**	90.10	**92.20**
GPT-4o	1.67	**3.11**	1.61	**3.20**	1.90	**3.77**	1.69	**3.69**	1.70	**2.32**
Human	2.10	**4.16**	1.94	**4.04**	2.38	**4.70**	1.74	**4.56**	2.30	**3.10**

Table 1: Evaluation of Sensor Data Trend Analysis Tasks for SensorLLM and GPT-4o. The assessment includes human and GPT-4o ratings (from 1 to 5, with 5 being the highest), as well as BLEU-1, ROUGE-1, ROUGE-L, METEOR, SBERT, and SimCSE (in %). The column *GPT-4o* refers to the trend analysis generated by GPT-4o itself, while the row *GPT-4o* refers to GPT-4o's evaluation of the generated outputs.

score accuracy and quality using the same criteria as GPT-4o, providing a human-centered perspective on the model's outputs.

Appendix A.5 details all metrics and scoring criteria. We randomly sample 200 instances per dataset for both SensorLLM and GPT-4o, then average the results for comparison. Because reading and comparing lengthy sequences is difficult for human annotators, we conduct human evaluation on 20 shorter sequences per dataset (each containing at most 50 time steps).

Results. Table 1 compares SensorLLM and GPT-4o on the Sensor Data Trend Analysis tasks, showing that our model consistently outperforms GPT-4o across all metrics. BLEU-1, ROUGE-1, ROUGE-L, and METEOR primarily focus on surface-level lexical or n-gram overlaps and SBERT and SimCSE can capture factual correctness or deeper semantic similarities. Across all metrics, SensorLLM generates trend descriptions more closely aligned with the ground truth. GPT-4o evaluations further highlight SensorLLM's superior ability to capture trend details and coherence, whereas GPT-4o struggles with complex numerical data and trend observations (Yehudai et al., 2024). Human evaluation also favors SensorLLM, particularly for shorter sequences. CAPTURE-24 results are weaker compared to other datasets, likely due to its longer sequences being trained with the same parameters. Overall, these findings validate the effectiveness of our Sensor-Language Alignment method in enhancing LLMs' ability to interpret complex numerical sequence. Appendix A.9 provides qualitative examples of outputs from both models.

4.3 Human Activity Recognition

Setup. In this section, we evaluate the performance of SensorLLM on HAR tasks. Each experiment runs for five trials, using 8 training epochs, a batch size of 4, gradient accumulation steps of 8, and a maximum sequence length of 4096. We report the F1 macro score A.8 to account for class imbalance across different activity categories.

Baselines. We benchmark SensorLLM against 11 baselines across two categories: (i) *TS models*—Transformer (Vaswani et al., 2017), Informer (Zhou et al., 2021), NS-Transformer (Liu et al., 2022), PatchTST (Nie et al., 2023), TimesNet (Wu et al., 2023a), and iTransformer (Liu et al., 2024c); (ii) *HAR models*—DeepConvLSTM (Ordóñez and Roggen, 2016), DeepConvLSTMAttn (Murahari and Plötz, 2018), and Attend (Abedin et al., 2021). We also include Chronos+MLP and GPT4TS (Zhou et al., 2023a) for a more comprehensive comparison. Full baseline details are in Appendix A.7.

Results. Table 2 reports the macro F1 scores (%) averaged over five random runs. SensorLLM achieves the highest performance on four datasets (USC-HAD, PAMAP2, MHealth, CAPTURE-24) and ranks second on UCI-HAR, demonstrating its effectiveness in handling diverse sensor data.

Notably, SensorLLM achieves a notable improvement on the CAPTURE-24 dataset, surpassing all baselines by a significant margin with a mean F1-macro score of 48.6%, which is 5.0% higher than Attend (43.6%). On USC-HAD, Sen-

Method	USC-HAD	UCI-HAR	PAMAP2	MHealth	CAPTURE-24
PatchTST	$45.2_{\pm 1.48}$	$86.8_{\pm 0.84}$	$82.0_{\pm 0.71}$	$80.0_{\pm 1.58}$	$35.6_{\pm 0.89}$
Ns-Transformer	$52.6_{\pm 2.30}$	$88.0_{\pm 0.71}$	$78.8_{\pm 0.84}$	$77.2_{\pm 1.48}$	$34.8_{\pm 1.10}$
Informer	$51.2_{\pm 1.30}$	$86.6_{\pm 1.14}$	$78.0_{\pm 1.58}$	$74.0_{\pm 0.71}$	$35.6_{\pm 0.55}$
Transformer	$49.6_{\pm 1.67}$	$85.4_{\pm 0.89}$	$77.0_{\pm 0.71}$	$75.2_{\pm 1.30}$	$32.8_{\pm 0.84}$
iTransformer	$48.4_{\pm 1.82}$	$81.8_{\pm 0.84}$	$76.6_{\pm 0.55}$	$80.4_{\pm 1.14}$	$19.8_{\pm 0.84}$
TimesNet	$52.2_{\pm 2.39}$	$87.4_{\pm 1.14}$	$76.2_{\pm 1.92}$	$78.4_{\pm 1.52}$	$34.8_{\pm 0.84}$
GPT4TS	$54.2_{\pm 2.05}$	$88.2_{\pm 0.84}$	$80.4_{\pm 0.89}$	$76.4_{\pm 1.14}$	$32.8_{\pm 1.10}$
Chronos+MLP	$44.2_{\pm 1.30}$	$82.2_{\pm 0.84}$	$79.8_{\pm 0.45}$	$83.0_{\pm 0.71}$	$38.0_{\pm 0.71}$
DeepConvLSTM	$48.8_{\pm 2.39}$	$89.2_{\pm 0.84}$	$78.4_{\pm 1.52}$	$75.0_{\pm 1.87}$	$40.4_{\pm 0.89}$
DeepConvLSTMAtt	$54.0_{\pm 2.12}$	$89.6_{\pm 1.14}$	$79.2_{\pm 1.30}$	$77.4_{\pm 2.19}$	$41.4_{\pm 0.55}$
Attend	$\underline{60.2}_{\pm 2.17}$	$\mathbf{93.2}_{\pm 0.84}$	$\underline{84.6}_{\pm 1.14}$	$\underline{83.4}_{\pm 1.14}$	$\underline{43.6}_{\pm 0.55}$
SensorLLM	$\mathbf{61.2}_{\pm 3.56}$	$\underline{91.2}_{\pm 1.48}$	$\mathbf{86.2}_{\pm 1.48}$	$\mathbf{89.4}_{\pm 3.85}$	$\mathbf{48.6}_{\pm 1.14}$

Table 2: F1-macro results (%) for the Task-Aware Tuning Stage, presented as the mean and standard deviation over 5 random repetitions. The top results for each dataset are highlighted as follows: **Bold** for the best and underline for the second-best.

Dataset	Task-only		SensorLLM	
	w/o prompts	w/ prompts	w/o prompts	w/ prompts
USC-HAD	$43.4_{\pm 2.88}$	$45.0_{\pm 1.58}$	$49.6_{\pm 1.67}$	$\mathbf{61.2}_{\pm \mathbf{3.56}}$
UCI-HAR	$80.0_{\pm 2.12}$	$82.0_{\pm 1.58}$	$89.2_{\pm 1.10}$	$\mathbf{91.2}_{\pm \mathbf{1.48}}$
PAMAP2	$74.2_{\pm 2.28}$	$75.4_{\pm 3.05}$	$83.0_{\pm 0.71}$	$\mathbf{86.2}_{\pm \mathbf{1.48}}$
MHealth	$76.6_{\pm 1.34}$	$77.4_{\pm 3.13}$	$86.6_{\pm 1.14}$	$\mathbf{89.4}_{\pm \mathbf{3.85}}$
CAPTURE-24	$44.8_{\pm 0.84}$	$46.0_{\pm 0.71}$	$47.2_{\pm 0.84}$	$\mathbf{48.6}_{\pm \mathbf{1.14}}$

Table 3: The results for SensorLLM trained with/without text prompts. *Task-only* refers to conducting HAR directly bypassing sensor-language alignment.

sorLLM achieves the highest score of 61.2%, outperforming Attend, the second-best baseline, by 1.0%. Similarly, on PAMAP2, SensorLLM achieves a score of 86.2%, exceeding Attend (84.6%) by 1.6%. On MHealth, SensorLLM sets a new state-of-the-art with a score of 89.4%, surpassing Attend (83.4%) by 6.0%. These results highlight SensorLLM's ability to consistently outperform existing methods across diverse datasets.

For UCI-HAR, SensorLLM achieves the second-best score (91.2%), slightly trailing Attend (93.2%). In contrast, Chronos+MLP shows only a slight improvement over iTransformer, the lowest-performing baseline (82.2% vs. 81.8%), indicating that Chronos embeddings alone have limited utility for HAR on this dataset. However, our framework significantly enhances their effectiveness, highlighting the robustness of our alignment approach.

5 Ablation Studies

Impact of Alignment. To evaluate the impact of alignment, we included the Chronos+MLP baseline in Section 4.3 to show that SensorLLM's performance is not solely driven by Chronos embeddings. Additionally, we compared SensorLLM with a Task-only model, which skips the Sensor-Language Alignment Stage and directly applies Chronos embeddings and the LLM for HAR. As shown in Table 3, SensorLLM consistently outperforms the Task-only model across all five datasets, regardless of additional textual input. Notably, the Task-only model often performs on par with or worse than traditional TS baselines, underscoring the importance of our alignment method. These results confirm that Chronos embeddings alone are insufficient for optimal HAR performance and that alignment is essential for enabling LLMs to effectively understand sensor data.

Impact of Prompts. To assess the role of additional textual information (e.g., statistical features for each sensor channel) in the Task-Aware Tuning Stage, we compared SensorLLM's performance with and without prompts. As shown in Table 3, incorporating prompts consistently improves F1-macro scores across all datasets, with a more pronounced effect in the full SensorLLM architecture. This demonstrates that the model effectively integrates sensor and textual data, enhancing its ability to capture complex temporal patterns. The results highlight the benefits of multimodal inputs, which enrich sensor data representations and improve HAR accuracy. More broadly, the ability to process both sensor signals and textual prompts not only enhances classification performance but also extends LLMs' potential for tackling complex

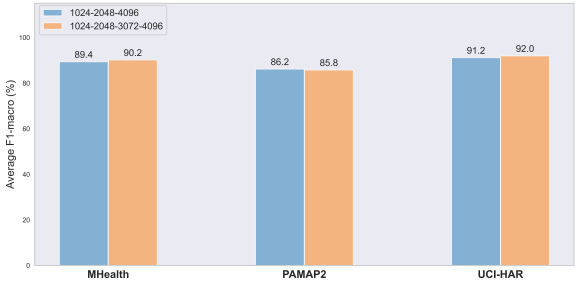

Figure 3: Effect of the number of alignment module layers.

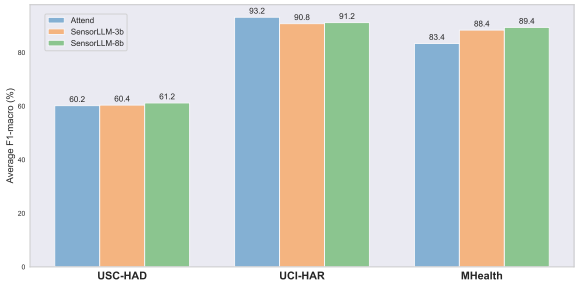

Figure 4: Effect of Model Size.

sensor-driven tasks in future applications.

Alignment Module Layers. We investigate the effect of hidden layer depth in the alignment module MLP on UCI-HAR, PAMAP2, and MHealth. As shown in Figure 3, increasing the number of hidden layers from one ($1024 \rightarrow 2048 \rightarrow 4096$) to two ($1024 \rightarrow 2048 \rightarrow 3072 \rightarrow 4096$) led to mixed results. The average F1-macro scores improved on UCI-HAR (91.2% $\rightarrow$ 92.0%) and MHealth (89.4% $\rightarrow$ 90.2%), but slightly dropped on PAMAP2 (86.2% $\rightarrow$ 85.8%). These results suggest that the optimal number of hidden layers varies by dataset.

Effect of Model Size on Performance. To assess the impact of model size, we tested SensorLLM-3b, a resource-efficient variant using Chronos-base and Llama3.2-3b. Experiments were conducted on USC-HAD, UCI-HAR, and MHealth. As shown in Figure 4, SensorLLM-3b achieves slightly lower performance than SensorLLM-8b across all datasets, illustrating the trade-off between model size and accuracy. However, SensorLLM-3b remains competitive, outperforming Attend on USC-HAD and MHealth, while trailing it only on UCI-HAR. These results indicate that SensorLLM-3b offers a resource-efficient alternative, maintaining strong performance relative to other baselines.

Cross-Dataset Generalization. To assess the robustness of SensorLLM, we conducted cross-dataset experiments, training the Sensor-Language Alignment Stage on USC-HAD and the Task-Aware Tuning Stage on UCI-HAR, and vice versa. These datasets share the same sensor channels but differ in sampling rates. As shown in Table 4, SensorLLM achieves performance comparable to training both stages on the same dataset. This indicates that once modality alignment is established, retraining on new datasets is unnecessary. SensorLLM effectively generalizes to downstream tasks across diverse datasets, demonstrating that the alignment stage enables the LLM to truly understand sensor data, rather than memorizing dataset-specific features. These results highlight SensorLLM's potential for broad cross-dataset generalization, laying the groundwork for future TS-LLM models.

Stage 1	Stage 2	Results
USC-HAD	UCI-HAR	**91.0**$_{\pm 1.41}$
UCI-HAR	USC-HAD	**61.6**$_{\pm 2.07}$

Table 4: Cross-dataset experiments.

6 Conclusions

We introduced SensorLLM, a multimodal framework that aligns sensor data with automatically generated text at a human-perception level, moving beyond machine-level alignment. SensorLLM effectively captures complex sensor patterns, achieving superior performance in HAR tasks. Experiments across diverse datasets demonstrate its robustness in handling variable-length sequences, multi-channel inputs, and textual metadata. Cross-dataset results further highlight its strong generalizability without requiring dataset-specific alignment. This work establishes a foundation for Sensor-Text MLLMs, with potential applications for sensor data analysis. We release our code and data generation pipeline to facilitate future research on integrating time-series and text, particularly in low-resource domains.

7 Limitations

While SensorLLM demonstrates strong performance in aligning sensor data with LLMs, certain limitations remain, offering directions for future exploration.

Classifier-Based Design. To ensure fair comparisons with existing HAR models (classification models), we adopt a classifier for downstream

tasks rather than fully exploiting the LLM's generative abilities. Although our results verify that the Sensor-Language Alignment Stage can generalize across datasets, relying on a fixed-class classifier may constrain the model's adaptability to new activity categories. Future work could explore using the LLM in a generative or prompt-based capacity, enabling broader application scenarios such as activity discovery or open-set recognition.

Scope of Sensor-Text Alignment. Our alignment focuses on mapping sensor data to trend-descriptive text, demonstrating clear benefits for LLM-based HAR. However, human-intuitive descriptions of sensor data extend beyond trend changes—incorporating frequency-domain features, periodicity, and higher-order patterns may further enhance an LLM's ability to interpret time-series data. Future research could investigate whether aligning text with alternative sensor characteristics improves time-series reasoning. This could expand the potential of multimodal NLP applications in sensor-driven tasks beyond activity recognition.

References

Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C. Ranasinghe. 2021. Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(1).

D. Anguita, Alessandro Ghio, L. Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *The European Symposium on Artificial Neural Networks*.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Oresti Baños, Rafael García, Juan Antonio Holgado Terriza, Miguel Damas, Héctor Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. 2014. mhealthdroid: A novel framework for agile development of mobile health applications. In *International Workshop on Ambient Assisted Living and Home Care*.

Antonio Bevilacqua, Kyle MacDonald, Aamina Rangarej, Venessa Widjaya, Brian Caulfield, and Tahar Kechadi. 2019. *Human Activity Recognition with Convolutional Neural Networks*, page 541–552. Springer International Publishing.

Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *Preprint*, arXiv:2310.04948.

Shing Chan, Hang Yuan, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. 2024. Capture-24: A large dataset of wrist-worn activity tracker data collected in the wild for human activity recognition. *Preprint*, arXiv:2402.19229.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2024. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *Preprint*, arXiv:2308.08469.

Shohreh Deldari, Dimitris Spathis, Mohammad Malekzadeh, Fahim Kawsar, Flora D. Salim, and Akhil Mathur. 2024. Crossl: Cross-modal self-supervised learning for time-series through latent masking. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 152–160.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. *Preprint*, arXiv:2310.07820.

Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(2).

Sojeong Ha and Seungjin Choi. 2016. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 381–388.

Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1533–1540. AAAI Press.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, and 5 others. 2021. Pre-trained models: Past, present and future. *Preprint*, arXiv:2106.07139.

Harish Haresamudram, David V. Anderson, and Thomas Plötz. 2019. On the role of features in human activity recognition. In *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ISWC '19, page 78–88, New York, NY, USA. Association for Computing Machinery.

Harish Haresamudram, Apoorva Beedu, Mashfiqui Rabbi, Sankalita Saha, Irfan Essa, and Thomas Ploetz. 2024. Limitations in employing natural language supervision for sensor-based human activity recognition–and ways to overcome them. *arXiv preprint arXiv:2408.12023*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Sijie Ji, Xinzhe Zheng, and Chenshu Wu. 2024. Hargpt: Are llms zero-shot human activity recognizers? *Preprint*, arXiv:2403.02727.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024a. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*.

Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, Shirui Pan, Vincent S. Tseng, Yu Zheng, Lei Chen, and Hui Xiong. 2023. Large models for time series and spatio-temporal data: A survey and outlook. *Preprint*, arXiv:2310.10196.

Ming Jin, Yifan Zhang, Wei Chen, Kexin Zhang, Yuxuan Liang, Bin Yang, Jindong Wang, Shirui Pan, and Qingsong Wen. 2024b. Position paper: What can large language models tell us about time series analysis. *Preprint*, arXiv:2402.02713.

Panagiotis Kasnesis, Charalampos Z. Patrikakis, and Iakovos S. Venieris. 2019. Perceptionnet: A deep convolutional neural network for late sensor fusion. In *Intelligent Systems and Applications*, pages 101–119, Cham. Springer International Publishing.

Yubin Kim, Xuhai Xu, Daniel McDuff, Cynthia Breazeal, and Hae Won Park. 2024. Health-llm: Large language models for health prediction via wearable sensor data. In *Proceedings of the fifth Conference on Health, Inference, and Learning*, volume 248 of *Proceedings of Machine Learning Research*, pages 522–539. PMLR.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82.

Hong Li, Gregory D. Abowd, and Thomas Plötz. 2018. On specialized window lengths and detector based human activity recognition. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, page 68–71, New York, NY, USA. Association for Computing Machinery.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Che Liu, Zhongwei Wan, Sibo Cheng, Mi Zhang, and Rossella Arcucci. 2024a. Etp: Learning transferable ecg representations via ecg-text pre-training. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8230–8234. IEEE.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning.

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

Xu Liu, Junfeng Hu, Yuan Li, Shizhe Diao, Yuxuan Liang, Bryan Hooi, and Roger Zimmermann. 2024b. Unitime: A language-empowered unified model for cross-domain time series forecasting. *Preprint*, arXiv:2310.09751.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024c. itransformer: Inverted transformers are effective for time series forecasting. *International Conference on Learning Representations*.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting.

Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Aparajita Saraf, Amy Bearman, and Babak Damavandi. 2023. IMU2CLIP: Language-grounded motion sensor translation with multimodal contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13246–13253, Singapore. Association for Computational Linguistics.

Vishvak S. Murahari and Thomas Plötz. 2018. On attention models for human activity recognition. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ISWC '18, page 100–103, New York, NY, USA. Association for Computing Machinery.

Shikai Qiu Nate Gruver, Marc Finzi and Andrew Gordon Wilson. 2023. Large Language Models Are Zero Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109.

Dimitris Spathis and Fahim Kawsar. 2024. The first step is the hardest: Pitfalls of representing and tokenizing temporal data for large language models. *Journal of the American Medical Informatics Association*, 31(9):2151–2158.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024a. Test: Text prototype aligned embedding to activate llm's ability for time series.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024b. Test: Text prototype aligned embedding to activate llm's ability for time series. *Preprint*, arXiv:2308.08241.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting? *Preprint*, arXiv:2406.16964.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023a. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023b. Next-gpt: Any-to-any multimodal llm.

Kang Xia, Wenzhong Li, Shiwei Gan, and Sanglu Lu. 2024. Ts2act: Few-shot human activity sensing with cross-modal co-learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4):1–22.

Qingxin Xia, Takuya Maekawa, and Takahiro Hara. 2023. Unsupervised human activity recognition through two-stage prompting with chatgpt. *Preprint*, arXiv:2306.02140.

Cheng Xu, Duo Chai, Jie He, Xiaotong Zhang, and Shihong Duan. 2019. Innohar: A deep neural network for complex human activity recognition. *IEEE Access*, 7:9893–9902.

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

Gilad Yehudai, Haim Kaplan, Asma Ghandeharioun, Mor Geva, and Amir Globerson. 2024. When can transformers count to n? *Preprint*, arXiv:2407.15160.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Hyungjun Yoon, Biniyam Aschalew Tolera, Taesik Gong, Kimin Lee, and Sung-Ju Lee. 2024. 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*, pages 2219–2241, Miami, Florida, USA. Association for Computational Linguistics.

Yuta Yuki, Junto Nozaki, Kei Hiroi, Katsuhiko Kaji, and Nobuo Kawaguchi. 2018. Activity recognition using dual-convlstm extracting local and global features for shl recognition challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, page 1643–1651, New York, NY, USA. Association for Computing Machinery.

Mi Zhang and Alexander A. Sawchuk. 2012. Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, page 1036–1043, New York, NY, USA. Association for Computing Machinery.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pages 11106–11115. AAAI Press.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023a. One Fits All: Power general time series analysis by pretrained lm. In *NeurIPS*.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023b. One fits all:power general time series analysis by pretrained lm. *Preprint*, arXiv:2302.11939.

Yunjiao Zhou, Jianfei Yang, Han Zou, and Lihua Xie. 2023c. Tent: Connect language models with iot sensors for zero-shot activity recognition. *arXiv preprint arXiv:2311.08245*.

A Appendix

A.1 More related work

Deep learning in human activity recognition. Over the last decade, HAR has transitioned from hand-crafted feature extraction to deep learning models capable of automatic feature learning. Early work by Kwapisz et al. (2011) utilized machine learning techniques, such as decision trees and MLPs, to classify activities using features extracted from wearable sensor data. Later, Haresamudram et al. (2019) demonstrated that optimized feature extraction within the Activity Recognition Chain (ARC) could rival or outperform end-to-end deep learning models. Deep learning models, particularly CNNs and LSTMs, have since become dominant in HAR. Bevilacqua et al. (2019) developed a CNN-based model for HAR, while Ha and Choi (2016) introduced CNN-pf and CNN-pff architectures that apply partial and full weight sharing for better feature extraction. Other notable works include Perception-Net Kasnesis et al. (2019), which leverages 2D convolutions for multimodal sensor data, and InnoHAR (Xu et al., 2019), which combines Inception CNN and GRUs for multiscale temporal feature learning. A dual-stream network utilizing convolutional layers and LSTM units, known as ConvLSTM, was employed by Yuki et al. (2018) to analyze complex temporal hierarchies with streams handling different time lengths. The combination of attention mechanisms with recurrent networks to enhance the computation of weights for hidden state outputs has also been demonstrated by DeepConvLSTM (Kasnesis et al., 2019) in capturing spatial-temporal features.

Large Language Models for Time-Series Forecasting. LLMs have achieved remarkable success in text-related tasks, and their utility has expanded into time-series forecasting. Xue and Salim (2023) presents PromptCast, which redefines time-series forecasting as a natural language generation task by transforming numerical inputs into textual prompts, enabling pre-trained language models to handle forecasting tasks with superior generalization in zero-shot settings. Gruver et al. (2023) explores encoding time-series as numerical strings, allowing LLMs like GPT-3 and LLaMA-2 to perform zero-shot forecasting, matching or surpassing the performance of specialized models, while highlighting challenges in uncertainty calibration due to model modifications like RLHF. Zhou et al. (2023b) demonstrates that pre-trained language and image models, such as a Frozen Pretrained Transformer (FPT), can be adapted for diverse time-series tasks like classification, forecasting, and anomaly detection, leveraging self-attention mechanisms to bridge the gap between different data types and achieving state-of-the-art performance across various tasks. Jin et al. (2024b) highlights the transformative potential of LLMs for time-series analysis by integrating language models with traditional analytical methods. Jin et al. (2024a) introduces a reprogramming framework that aligns time-series data with natural language processing capabilities, enabling LLMs to perform time-series forecasting without altering the core model structure. Cao et al. (2024) presents TEMPO, a generative transformer framework based on prompt tuning, which adapts pre-trained models for time-series forecasting by decomposing trends, seasonality, and residual information. Sun et al. (2024b) proposes TEST, an innovative embedding technique that integrates time-series data with LLMs through instance-wise, feature-wise, and text-prototype-aligned contrast, yielding improved or comparable results across various applications. Chang et al. (2024) develops a framework that enhances pre-trained LLMs for multivariate time-series forecasting through a two-stage fine-tuning process and a novel multi-scale temporal aggregation method, outperforming traditional models in both full-shot and few-shot scenarios. Finally, Liu et al. (2024b) introduces UniTime, a unified model that leverages language instructions and a Language-TS Transformer to handle multivariate time series across different domains, demonstrating enhanced forecasting performance and zero-shot transferability.

LLMs for Human Activity Recognition. While LLMs like ChatGPT have demonstrated remarkable performance in various NLP tasks, their effectiveness in HAR remains limited due to challenges in interpreting sensor data. These models often struggle to distinguish between activities that share similar objects, requiring more advanced prompt engineering to highlight activity-specific details. (Xia et al., 2023) proposed an unsupervised approach to HAR using ChatGPT, leveraging two-stage prompts to infer activities from object sequences without manual descriptions. The method demonstrates superior performance on three benchmark datasets, marking a significant advancement in applying language models to activity recognition

tasks. Similarly, Ji et al. (2024) explored LLMs for zero-shot HAR using raw IMU data, showing that GPT-4 can outperform both traditional and deep learning models in simple HAR tasks without domain-specific adaptations, highlighting LLMs' potential in sensor-based systems.

A.2 Data Generation

We generate text data from sensor readings using predefined sentence templates (5, Tables 6,, 7). These templates are randomly selected to create diverse question-answer (QA) pairs. To enhance variability, we employ GPT-4o to generate synonymous variations. Each sentence contains placeholders for numerical values (e.g., timestamps, sensor readings) or textual information, which are dynamically replaced to produce coherent QA pairs aligned with the sensor data.

Trend Description Templates

- {start_time}s to {end_time}s: {trend}
- {start_time} seconds to {end_time} seconds: {trend}
- {start_time} to {end_time} seconds: {trend}
- {start_time}-{end_time} seconds: {trend}
- {start_time}-{end_time}s: {trend}
- {start_time}s-{end_time}s: {trend}

Table 5: Examples of answer templates used for trend descriptions.

The system prompt instructs the model on how to respond to generated questions, incorporating dataset-specific attributes such as sensor frequency and sampling rate. These tailored prompts ensure responses align with the unique characteristics of each dataset. Below is the system prompt template used for all datasets:

- A dialogue between a researcher and an AI assistant. The AI analyzes a sensor time-series dataset (N points, sampled at {sample_rate}Hz) to answer specific questions, demonstrating its analytical capabilities and the potential for human-AI collaboration in interpreting sensor data.

A.3 Chronos

Chronos (Ansari et al., 2024) is a pretrained probabilistic time-series framework that tokenizes real-valued time-series data into discrete representations for language model training. It utilizes scaling and quantization to transform time-series data into a fixed vocabulary, enabling T5-based (Raffel et al., 2020) models to learn from tokenized sequences using cross-entropy loss. Pretrained on diverse public and synthetic datasets, Chronos surpasses existing models on familiar datasets and demonstrates strong zero-shot performance on unseen tasks, making it a versatile tool for time-series forecasting across domains.

Time-Series Tokenization and Quantization. Chronos converts time-series data into discrete tokens through a two-step process: normalization and quantization. Mean scaling is first applied to ensure consistency across different time series:

$$\tilde{x} = \frac{x}{\text{mean}(|x|)} \tag{5}$$

Next, the normalized values are quantized using B bin centers $c_1, \ldots, c_B$ and corresponding bin edges $b_1, \ldots, b_{B-1}$, mapping real values to discrete tokens via:

$$q(x) = \begin{cases} 1 & \text{if } -\infty \leq x < b_1, \\ 2 & \text{if } b_1 \leq x < b_2, \\ \vdots & \\ B & \text{if } b_{B-1} \leq x < \infty. \end{cases} \tag{6}$$

Special tokens such as `PAD` and `EOS` are added to handle sequence padding and denote the end of sequences, allowing Chronos to process variable-length inputs efficiently within language models.

Objective Function. Chronos models the tokenized time series using a categorical distribution over the vocabulary V_{ts}, minimizing the cross-entropy loss:

$$\ell(\theta) = -\sum_{h=1}^{H+1} \sum_{i=1}^{|V_{ts}|} \mathbf{1}(z_{C+h+1} = i) \cdot \log p_\theta(z_{C+h+1} = i \mid z_{1:C+h}) \tag{7}$$

where C is the historical context length, H is the forecast horizon, and p_θ is the predicted token distribution.

Trend Description Templates

- Kindly provide a detailed analysis of the trend changes observed in the {data}.
- Please offer a comprehensive description of how the trends in the {data} have evolved.
- I would appreciate a thorough explanation of the trend fluctuations that occurred within the {data}.
- Could you examine the {data} in depth and explain the trend shifts observed step by step?
- Detail the {data}'s trend transitions.
- Could you assess the {data} and describe the trend transformations step by step?
- Could you analyze the trends observed in the {data} over the specified period step by step?
- Can you dissect the {data} and explain the trend changes in a detailed manner?
- What trend changes can be seen in the {data}?

Summary Templates

- Could you provide a summary of the main features of the input {data} and the distribution of the trends?
- Please give an overview of the essential attributes of the input {data} and the spread of the trends.
- Describe the salient features and trend distribution within the {data}.
- Give a summary of the {data}'s main elements and trend apportionment.
- Summarize the {data}'s core features and trend dissemination.
- Outline the principal aspects and trend allocation of the {data}.
- Summarize the key features and trend distribution of the {data}.
- I need a summary of {data}'s main elements and their trend distributions.

Table 6: Examples of question templates used for trend description and summary generation.

Summary 1: Trend Count

- Number of {trend} trends: {num}
- Count of {trend} trends: {num}
- Number of {trend} segments: {num}
- Count of {trend} segments: {num}

Summary 2: Sensor Data Context

- The given {data_name} represents {sensor_name} sensor readings from {start_time}s to {end_time}s.
- The {data_name} contains {sensor_name} sensor readings recorded between {start_time} and {end_time} seconds.
- The {sensor_name} sensor readings collected from {start_time} to {end_time} seconds are presented in this {data_name}.

Summary 3: Trend Change Statistics

- The data exhibits {trend_num} distinct trends, with {change_num} trend changes observed.
- Across {trend_num} trends, the data shows {change_num} occurrences of trend shifts.
- {trend_num} trends are present, with {change_num} instances of trend changes.

Summary 4: Cumulative Trend Analysis

- To sum up, the data exhibited a {trend_type} trend for a total duration of {total_time} seconds.
- Overall, the data showed a {trend_type} trend spanning {total_time} seconds.
- In conclusion, the trend was {trend_type} over {total_time} seconds.

Summary 5: Overall Trend Summary

- The overall trend is {overall_trend}.
- The primary trend detected is {overall_trend}.
- Looking at the broader pattern, the trend is {overall_trend}.

Table 7: Examples of answer templates used for summaries.

This approach offers two key advantages: (i) Seamless integration with language models, requiring no architectural modifications, and (ii) Flexible distribution learning, enabling robust generalization across diverse time-series datasets.

A.4 GPT-4o Prompt for Sensor Data Trend Analysis

Table 8 presents the system prompt used to generate trend-descriptive texts from sensor data, providing a structured framework for GPT-4o to analyze and respond to specific questions. This standardized prompt ensures consistency in GPT-4o's interpretation of time-series data, allowing direct comparison with descriptions produced by SensorLLM.

Prompt	A dialogue between a curious researcher and an AI assistant. The AI analyzes a sensor time-series dataset (N points, {sr}Hz sampling rate) to answer specific questions. Please output your answer in the format like this example: {example from ground-truth} Now, analyze the following: Input: {sensor_data} How trends in the given sensor data evolve? Output:

Table 8: Prompt for GPT-4o to generate descriptive texts based on the given numerical sensor data.

We evaluate GPT-4o's ability to interpret numerical sensor data by assessing its responses against human evaluations and NLP metrics. This comparison benchmarks GPT-4o's performance against SensorLLM, highlighting differences in how both models process time-series data trends. The results demonstrate the effectiveness of SensorLLM's Sensor-Language Alignment Stage.

A.5 Evaluation Metrics for Sensor-Language Alignment Stage

In this section, we describe the various evaluation metrics used to assess the performance of SensorLLM in generating trend descriptions from sensor data. Each metric offers a distinct perspective on model performance, ranging from surface-level textual similarity to more complex semantic alignment.

BLEU-1 (Papineni et al., 2002). BLEU (Bilingual Evaluation Understudy) is a precision-based metric commonly used to evaluate machine-generated text by comparing it to reference texts. BLEU-1 focuses on unigram (single-word) overlap, assessing the lexical similarity between the generated and reference text. While useful for measuring word-level matches, BLEU-1 does not capture deeper semantic meaning, making it most effective for surface-level alignment.

ROUGE-1 and ROUGE-L (Lin, 2004). ROUGE (Recall-Oriented Understudy for Gisting Evaluation) evaluates the recall-oriented overlap between generated text and reference text. ROUGE-1 focuses on unigram recall, similar to BLEU-1 but emphasizing how much of the reference text is captured. ROUGE-L measures the longest common subsequence, assessing both precision and recall in terms of structure and content overlap, though it does not evaluate semantic accuracy.

METEOR (Banerjee and Lavie, 2005). METEOR (Metric for Evaluation of Translation with Explicit Ordering)combines precision and recall, with additional alignment techniques such as stemming and synonym matching. Unlike BLEU and ROUGE, METEOR accounts for some degree of semantic similarity. However, its emphasis is still on word-level alignment rather than factual accuracy or meaning.

SBERT (Reimers and Gurevych, 2019). SBERT (Sentence-BERT) [2] is a metric that generates sentence embeddings using the BERT architecture. It computes cosine similarity between embeddings of the generated and reference texts, providing a deeper assessment of semantic similarity beyond lexical matches.

SimCSE (Gao et al., 2021). SimCSE (Simple Contrastive Sentence Embedding) [3] introduces a contrastive learning approach to fine-tune language models for sentence embeddings. By applying different dropout masks to the same sentence, it generates positive examples, encouraging similar embeddings for semantically identical sentences while distinguishing different ones.

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://huggingface.co/princeton-nlp/sup-simcse-roberta-large

GPT-4o Evaluation. In addition to the NLP metrics, we also employed GPT-4o as a human-like evaluator. Given its strong reasoning and comprehension abilities, GPT-4o was tasked with scoring the generated text based on its alignment with the ground truth. GPT-4o evaluated the correctness, completeness, and coherence of the trend descriptions and assigned a score from 1 to 5, accompanied by an explanation (see Table 9). This type of evaluation provides insights into how well the generated outputs capture the nuances of sensor data trends in a manner similar to human understanding.

Human Evaluation. Finally, five human experts assessed the correctness and quality of the generated trend descriptions. Following the same criteria as GPT-4o, they rated the outputs on a scale from 1 to 5, focusing on the factual accuracy and coherence of the descriptions. This manual evaluation serves as an important benchmark for the model's performance from a human perspective, ensuring that the generated outputs are not only technically correct but also practically useful for human interpretation.

A.6 Datasets

We used five datasets in our study:

USC Human Activity Dataset (USC-HAD). USC-HAD (Zhang and Sawchuk, 2012) consists of six sensor readings from body-worn 3-axis accelerometers and gyroscopes, collected from 14 subjects. The data is sampled at 100 Hz across six channels and includes 12 activity class labels. For evaluation, we use data from subjects 13 and 14 as the test set, while the remaining subjects' data are used for training. A window size $w \in [5, 200]$ is used in alignment stage, and $w = 200$ with stride of 100 are used in HAR.

UCI Human Activity Recognition Dataset (UCI-HAR). UCI-HAR (Anguita et al., 2013) includes data collected from 30 volunteers performing six activities while wearing a smartphone on their waist. The embedded accelerometer and gyroscope sensors sampled data at 50 Hz across six channels. The dataset was partitioned into 70% for training and 30% for testing. A window size $w \in [5, 200]$ is used in alignment stage, and $w = 128$ with stride of 64 is used in HAR.

Physical Activity Monitoring Dataset (PAMAP2). PAMAP2 (Reiss and Stricker, 2012) includes data from nine subjects wearing IMUs on their chest, hands, and ankles. IMUs capture the acceleration, gyroscope, and magnetometer data across 27 channels and include 12 activity class labels. For our experiments, data from subjects 105 and 106 are used as the test set, with the remaining subjects' data used for training. The sample rate is downsampled from 100 Hz to 50 Hz. A window size $w \in [5, 100]$ is used in alignment stage, and $w = 100$ with stride of 50 in HAR.

Mobile Health Dataset (MHealth). MHealth (Baños et al., 2014) contains body motion and vital sign recordings from ten volunteers. Sensors were placed on the chest, right wrist, and left ankle of each subject. For our experiments, we used acceleration data from the chest, left ankle, and right lower arm, along with gyroscope data from the left ankle and right lower arm, resulting in a total of 15 channels. The data is sampled at 50 Hz and includes 12 activity class labels. Data from subjects 1, 3, and 6 is used as the test set, while the remaining subjects' data are used for training. We use a window size $w \in [5, 100]$ in alignment stage and $w = 100$ with stride of 50 in HAR.

CAPTURE-24. CAPTURE-24 (Chan et al., 2024) is a large-scale dataset featuring 3-channel wrist-worn accelerometer data collected in free-living settings for over 24 hours per participant. It includes annotated data from 151 participants, making it significantly larger than existing datasets. We used the first 100 participants as the training set and the remaining 51 as the test set. For each subject, sequences were windowed, and 5% of the data was randomly selected for training and testing. The sample rate was downsampled from 100 Hz to 50 Hz and it includes 10 activity class labels. During the alignment stage, we used a variable window size $w \in [10, 500]$, while in the HAR, we fixed $w = 500$ with a stride of 250.

Each dataset includes multiple activity classes, and the proportion of each class in the dataset is shown in Table 10.

A.7 Baselines for Task-Aware Tuning Stage

In Task-Aware Tuning Stage, we compare SensorLLM against several state-of-the-art baseline models for time-series classification and human activity recognition (HAR). These models were selected for their strong performance in relevant tasks, providing a thorough benchmark for evaluating SensorLLM's effectiveness.

Prompt	Please evaluate the model-generated trend descriptions against the ground truth. Rate each pair based on the degree of accuracy, using a scale from 1 to 5, where 1 represents the lowest correctness and 5 represents the highest. Deduct 1 point for minor errors in the trend description, and 2-3 points for moderate errors. Provide your score (1-5) and a brief explanation in the format: "score#reason" (e.g., 4#The description of trend changes slightly differs from the ground truth). Now, please proceed to score the following: Model: {model_output} Human: {ground_truth} Output:
Output example 1:	2#Significant discrepancies in segment durations and trend counts compared to ground-truth.
Output example 2:	5#The model's description matches the human-generated text accurately.

Table 9: Prompt and output examples for GPT-4o in evaluating model-generated texts and ground-truth.

Dataset	# Classes	Classes	Proportions (%)
USC-HAD	12	Sleeping, Sitting, Elevator down, Elevator up, Standing, Jumping, Walking downstairs, Walking right, Walking forward, Running forward, Walking upstairs, Walking left	12.97, 9.06, 6.04, 5.94, 8.6, 3.62, 7.61, 9.81, 13.15, 5.72, 8.22, 9.25
UCI-HAR	6	Standing, Sitting, Laying, Walking, Walking downstairs, Walking upstairs	18.69, 17.49, 19.14, 16.68, 13.41, 14.59
PAMAP2	12	Lying, Sitting, Standing, Ironing, Vacuum cleaning, Ascending stairs, Descending stairs, Walking, Nordic walking, Cycling, Running, Rope jumping	10.25, 9.52, 10.11, 11.82, 9.14, 6.3, 5.67, 12.77, 9.52, 8.42, 3.57, 2.91
MHealth	12	Climbing stairs, Standing still, Sitting and relaxing, Lying down, Walking, Waist bends forward, Frontal elevation of arms, Knees bending (crouching), Jogging, Running, Jump front & back, Cycling	8.91, 8.95, 8.95, 8.95, 8.95, 8.26, 8.7, 8.53, 8.95, 8.95, 2.96, 8.95
CAPTURE-24	10	Sleep, Household-chores, Walking, Vehicle, Standing, Mixed-activity, Sitting, Bicycling, Sports, Manual-work	37.45, 6.5, 6.16, 3.83, 3.25, 3.49, 37.07, 1.03, 0.43, 0.79

Table 10: Dataset classes and Proportions

Transformer (Vaswani et al., 2017). The Transformer model is a widely-used architecture in various tasks, including time-series forecasting and classification. It uses self-attention mechanisms to capture long-range dependencies in sequential data, making it highly effective for modeling complex temporal relationships.

Informer (Zhou et al., 2021). Informer is a transformer-based model designed for long sequence time-series data. It addresses key limitations of standard Transformers, such as high time complexity and memory usage, through three innovations: ProbSparse self-attention, which reduces time complexity; self-attention distilling, which enhances efficiency by focusing on dominant patterns; and a generative decoder that predicts entire sequences in a single forward pass.

NS-Transformer (Liu et al., 2022). Non-stationary Transformers (NS-Transformer) tackles the issue of over-stationarization in time-series by balancing series predictability and model capability. It introduces Series Stationarization to normalize inputs and De-stationary Attention to restore intrinsic non-stationary information into temporal dependencies.

PatchTST (Nie et al., 2023). PatchTST is a Transformer-based model for multivariate time series tasks, using subseries-level patches as input tokens and a channel-independent approach to reduce computation and improve efficiency. This design retains local semantics and allows for longer historical context, significantly improving long-term forecasting accuracy.

TimesNet (Wu et al., 2023a). TimesNet is a versatile backbone for time series analysis that transforms 1D time series into 2D tensors to better capture intraperiod and interperiod variations. This 2D transformation allows for more efficient modeling using 2D kernels. It also introduces TimesBlock to adaptively discovers multi-periodicity and extracts temporal features from transformed 2D tensors using a parameter-efficient inception block.

iTransformer (Liu et al., 2024c). iTransformer reimagines the Transformer architecture by applying attention and feed-forward networks to inverted dimensions. Time points of individual series are embedded as variate tokens, allowing the attention mechanism to capture multivariate correlations, while the feed-forward network learns nonlinear representations for each token.

DeepConvLSTM (Ordóñez and Roggen, 2016). DeepConvLSTM integrates four consecutive convolutional layers followed by two LSTM layers to effectively capture both spatial and temporal dynamics in sensor data. The final output vector is passed through a fully connected layer, and the softmax function is applied to produce activity class probabilities as the model's final output.

DeepConvLSTMAttn (Murahari and Plötz, 2018). DeepConvLSTMAttn enhances the original DeepConvLSTM by integrating an attention mechanism to improve temporal modeling in HAR tasks. Instead of using the last LSTM hidden state for classification, the attention mechanism is applied to the first 7 hidden states, representing historical temporal context. These states are transformed through linear layers to generate attention scores, which are passed through softmax to produce weights. The weighted sum of the hidden states is combined with the last hidden state to form the final embedding for classification.

Attend (Abedin et al., 2021). The Attend model use the latent relationships between multi-channel sensor modalities and specific activities, apply data-agnostic augmentation to regularize sensor data streams, and incorporate a classification loss criterion to minimize intra-class representation differences while maximizing inter-class separability. These innovations result in more discriminative activity representations, significantly improving HAR performance.

Chronos+MLP. Chronos (Ansari et al., 2024)+MLP is a baseline designed to evaluate whether the performance gains in SensorLLM are solely attributable to Chronos and the MLP. In SensorLLM, Chronos is used to generate sensor embeddings, which are then mapped by the MLP for input into the LLM to perform HAR. Since Chronos does not natively support classification tasks and only processes single-channel data, we adapt it for HAR by inputting each channel's data separately into Chronos. The resulting sensor embeddings for all channels are then concatenated and fed into an MLP, which acts as a classifier. This setup allows us to benchmark against a simpler framework and validate the unique contributions of SensorLLM's design.

GPT4TS (Zhou et al., 2023a). GPT4TS is a unified framework that leverages a frozen pre-trained language model (e.g., GPT-2 (Radford et al., 2019)) to achieve state-of-the-art or comparable performance across various time-series analysis tasks, including classification, forecasting (short/long-term), imputation, anomaly detection, and few-shot/zero-sample forecasting. The authors also found that self-attention functions similarly to PCA, providing a theoretical explanation for the versatility of transformers.

A.8 Evaluation Metrics for Task-Aware Tuning Stage

In our evaluation, we use the F1-macro score to assess the model's performance across datasets. F1-macro is particularly suitable for datasets with imbalanced label distributions, which is common in Human Activity Recognition (HAR) tasks where certain activities are overrepresented while others have fewer samples. Unlike the micro F1 score, which emphasizes the performance on frequent classes, F1-macro treats each class equally by calculating the F1 score independently for each class and then averaging them.

The formula for the F1-macro score is:

$$\text{F1-macro} = \frac{1}{C} \sum_{i=1}^{C} \text{F1}_i \tag{8}$$

where C is the total number of classes, and F1_i is the F1 score for class i. The F1 score for each class is calculated as:

$$\text{F1}_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{9}$$

The precision and recall for each class are defined as:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{10}$$

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{11}$$

where TP_i, FP_i, and FN_i represent the number of true positives, false positives, and false negatives for class i, respectively. This metric ensures that performance is evaluated fairly across all classes, regardless of the frequency of each label, making it a robust measure for imbalanced datasets.

A.9 Sensor-Language Alignment Stage Output Examples

Tables 11 and 12 present two examples of the trend analysis results generated by SensorLLM and GPT-4o based on the input sensor data. From the results, it is evident that SensorLLM outperforms GPT-4o across both shorter and medium-length sequences. This demonstrates that our approach enables LLMs to better understand numerical variations, as well as accurately compute the time duration represented by the input sequences based on their length and the given sample rate. In contrast, current large language models struggle with directly interpreting numerical data, as their tokenization methods are not well-suited for tasks such as comparing numerical values or counting (Yehudai et al., 2024).

Sensor readings:	[-9.8237, -9.4551, -10.007, -11.273, -11.258, -11.677, -11.774, -11.638, -11.195, -11.087, -10.833, -11.044, -11.393, -11.943, -12.168, -15.455, -12.967, -12.326, -12.515, -13.195, -12.634, -11.873, -12.002, -11.583, -10.859, -10.349, -9.831, -9.1622, -8.2721, -6.9299, -6.255, -5.5998]
Ground-truth:	0.0 seconds to 0.02 seconds: growing 0.02 seconds to 0.06 seconds: declining 0.06 seconds to 0.08 seconds: growing 0.08 seconds to 0.12 seconds: declining 0.12 seconds to 0.2 seconds: growing 0.2 seconds to 0.3 seconds: declining 0.3 seconds to 0.34 seconds: growing 0.34 seconds to 0.38 seconds: declining 0.38 seconds to 0.42 seconds: growing 0.42 seconds to 0.44 seconds: declining 0.44 seconds to 0.62 seconds: growing Total growing trends: 6 Total declining trends: 5 From 0.0s to 0.62s, normalized left-ankle y-axis accelerometer data is showcased in the sensor data. Examining the data, we notice 2 clear trend characteristics, with the trend fluctuating a total of eleven times. The analysis reveals that the data's declining inclination persisted for a total of 0.24 seconds, and a growing trend within a span of 0.38 seconds. The general trend observed is growing.
SensorLLM:	0.0 to 0.02 seconds: ascending 0.02 to 0.08 seconds: descending 0.08 to 0.1 seconds: ascending 0.1 to 0.12 seconds: descending 0.12 to 0.2 seconds: ascending 0.2 to 0.28 seconds: descending 0.28 to 0.32 seconds: ascending 0.32 to 0.36 seconds: descending 0.36 to 0.4 seconds: ascending 0.4 to 0.42 seconds: descending 0.42 to 0.62 seconds: ascending Count of ascending segments: 6 Count of descending segments: 5 The time series data encapsulates normalized left-ankle y-axis accelerometer sensor readings from 0.0 seconds to 0.62 seconds. Two separate trends and nine trend shifts are observed in the data. The analysis reveals that the data's descending inclination persisted for a total of 0.22 seconds, and an ascending trend for a sum of 0.40 seconds. The trend overall is ascending.
GPT-4o Evaluation:	3#Moderate errors in trend durations and shifts compared to ground truth.
GPT-4o's output:	0.0s to 0.02s: rising 0.02s to 0.06s: falling 0.06s to 0.08s: rising

	0.08s to 0.1s: falling 0.1s to 0.12s: rising 0.12s to 0.14s: falling 0.14s to 0.3s: rising Total rising segments: 4 Total falling segments: 3 The normalized left-ankle y-axis accelerometer sensor readings recorded within the 0.0 to 0.3 second timeframe are presented in this sensor data. The input data displays three individual trends, with a comprehensive change count reaching 7. The examination reveals that the data's falling inclination endured for an aggregate of 0.08 seconds, succeeded by a rising trend for a cumulative period of 0.22 seconds, and a steady pattern for a total of 0.00 seconds. The dominant trend is rising.
GPT-4o Evaluation:	2#Significant discrepancies in trend durations and counts between model and ground-truth.

Table 11: Example 1 of SensorLLM and GPT-4o Generated Descriptions with Ground-Truth and GPT-4o Evaluation

Sensor readings:	[0.53137, 0.53137, 0.53137, 0.51176, 0.51176, 0.51176, 0.45098, 0.45098, 0.45098, 0.45098, 0.45882, 0.45882, 0.45882]
Ground-truth:	0.0 seconds to 0.04 seconds: stable 0.04 seconds to 0.06 seconds: decreasing 0.06 seconds to 0.1 seconds: stable 0.1 seconds to 0.12 seconds: decreasing 0.12 seconds to 0.18 seconds: stable 0.18 seconds to 0.2 seconds: increasing 0.2 seconds to 0.24 seconds: stable Number of stable trends: 4 Number of decreasing trends: 2 Number of increasing trends: 1 The sensor data represents readings taken from a normalized right-lower-arm x-axis gyroscope sensor between 0.0 and 0.24 seconds. Analysis reveals three separate trends within the data, undergoing a cumulative total of seven shifts in direction. Encapsulating the outcomes, the data's decreasing trend stretched across a total time of 0.04 seconds, came after an increasing pattern observed over 0.02 seconds, and a stable trend for 0.18 seconds in total. The dominant trend is decreasing.
SensorLLM:	0.0s to 0.04s: consistent 0.04s to 0.06s: downward 0.06s to 0.1s: consistent 0.1s to 0.12s: downward 0.12s to 0.18s: consistent 0.18s to 0.2s: upward 0.2s to 0.24s: consistent

	Number of consistent segments: 4 Number of downward segments: 2 Number of upward segments: 1 The sensor data illustrates normalized right-lower-arm x-axis gyroscope sensor readings between 0.0 and 0.24 seconds. The input data displays three individual trends, with a comprehensive change count reaching 7. To encapsulate, the data's downward trend spanned a combined duration of 0.04 seconds, and then an upward pattern for a sum of 0.02 seconds, and a consistent trend for an accumulated time of 0.18 seconds. The overarching trend is characterized as downward.
GPT-4o Evaluation:	5#The model's description matches the ground truth accurately.
GPT-4o's output:	0.0s to 0.06s: steady 0.06s to 0.12s: falling 0.12s to 0.14s: steady 0.14s to 0.16s: rising 0.16s to 0.18s: steady Total steady segments: 3 Total rising segments: 1 Total falling segments: 1 The normalized right-lower-arm x-axis gyroscope sensor readings recorded within the 0.0 to 0.18 second timeframe are presented in this sensor data. The input data displays three individual trends, with a comprehensive change count reaching 5. The examination reveals that the data's falling inclination endured for an aggregate of 0.06 seconds, succeeded by a rising trend for a cumulative period of 0.02 seconds, and a steady pattern for a total of 0.10 seconds. The dominant trend is steady.
GPT-4o Evaluation:	2#Significant discrepancies in trend durations and counts compared to ground-truth.

Table 12: Example 2 of SensorLLM and GPT-4o Generated Descriptions with Ground-Truth and GPT-4o Evaluation