LEARNING TRANSFERABLE SENSOR MODELS VIA LANGUAGE-INFORMED PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal language models have demonstrated strong semantic understanding and reasoning over physiological and behavioral signals captured from diverse healthcare sensors. However, existing sensor-language models can only process a fixed number of sensor channels at a fixed sampling rate, making them difficult to adapt to new or different sensor configurations. This inflexibility prevents models from generalizing across heterogeneous health sensing ecosystems, where data spans high-frequency wearable streams to sparse, day-scale clinical signals. To bridge this gap, we introduce **SLIP** (**S**ensor **L**anguage-**I**nformed **P**retraining), an open-source framework for learning language-aligned representations that generalize across diverse sensor setups. SLIP integrates contrastive alignment with sensor-conditioned captioning, facilitating both discriminative understanding and generative reasoning. By repurposing a pretrained decoder-only language model via cross-attention and introducing a flexible patch embedder, SLIP transfers to new sensor configurations at inference without retraining, regardless of their native sampling rate or input length. Across 11 datasets, SLIP demonstrates superior performance in retrieval, signal captioning, and question answering. It achieves a 77.14% average linear-probing accuracy, a 3.56% improvement over the strongest baseline (SensorLM). Beyond classification, SLIP supports open-vocabulary sensor captioning and question answering without any architectural modifications. Our work offers broad implications for the development of sensor-language models, utilizing cross-modal supervision to ensure more robust generalizability in health sensing applications.[1]

1 INTRODUCTION

Healthcare relies on a diverse array of sensors, from wearables to clinical monitors, which capture longitudinal physiological and behavioral data critical for automated diagnosis and intervention. To enable scalable analysis of these signals, self-supervised learning (SSL) has become the dominant pretraining paradigm for learning general-purpose representations from unlabeled sensor data. Recent work shows that such pretraining improves a range of health applications spanning activity and sleep monitoring, stress recognition, and clinical diagnosis (Spathis et al., 2022; Thapa et al., 2026; Li et al., 2025). However, because sensor modalities differ fundamentally in sampling rate, channel structure, and signal dynamics, many such efforts are restricted to a single modality such as ECG or PPG (Pillai et al., 2024; Saha et al., 2025; McKeen et al., 2025), achieving strong in-domain performance but failing to generalize across the heterogeneous sensor configurations encountered in real-world deployments. NormWear (Luo et al., 2025) handles an arbitrary number of channels and sensor modalities, but requires a fixed sampling rate and input length. In parallel, general-purpose time series foundation models (Ansari et al., 2025; Goswami et al., 2024) pretrain on mixture corpora spanning multiple sensor modalities with predictive SSL objectives. However, these models are optimized for forecasting, and whether their learned representations transfer to downstream classification in domains such as human activity recognition, clinical diagnosis, and physiological monitoring remains largely unexplored (Figure 1).

Recently, large language models (LLMs) have showcased strong semantic abstraction and generalization in zero-shot and few-shot settings. Consequently, recent work explores aligning LLMs

[1]We plan to release our model weights, data, and code to support further research.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1: **Illustrated example of the forecasting-classification gap**. Chronos-2 achieves low reconstruction error on UCI-HAR (MSE = 0.96), yet its latent representations fail to discriminate between semantically distinct activities (walking downstairs vs. upstairs). This shows that predictive SSL does not necessarily yield representations that transfer to downstream classification tasks.

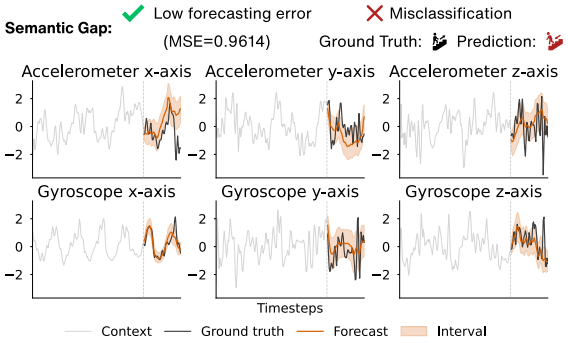


Table 1: **Comparison of sensor-text modeling approaches.** We compare prior work and SLIP across key capabilities, including **length-adaptive** encoding (patch size is adjusted at inference to match the input length and sampling rate, requiring no retraining), **sensor-agnostic** input (no fixed sensor set required), cross-modal **retrieval**, and sensor **question answering (QA)**.

Study	Length Adaptive	Sensor Agnostic	Retrieval	QA
Chronos (Ansari et al., 2024)	✗	✓	✗	✗
Chronos2 (Ansari et al., 2025)	✗	✓	✗	✗
Sundial (Liu et al., 2025)	✗	✓	✗	✗
SensorLM (Zhang et al., 2025)	✗	✗	✓	✗
NormWear (Luo et al., 2025)	✗	✓	✓	✗
ChatTS (Xie et al., 2024)	✗	✓	✗	✓
ITFormer (Wang et al., 2025)	✗	✗	✗	✓
OpenTSLM (Langer et al., 2025)	✗	✗	✗	✓
SLIP (ours)	✓	✓	✓	✓

with time series and sensor representations to overcome the limitations of reconstruction-based or sensor-specific pretraining. For instance, HealthLLM (Kim et al., 2024) reformulates sensor signals as text to leverage pretrained LLMs; however, this often leads to substantial semantic and temporal information loss. To better preserve signal structure, recent methods map time series into the LLM embedding space via projection layers and rely on the language model for contextual inference (Jin et al., 2023; Xie et al., 2024; Wang et al., 2025). A complementary line of work uses contrastive cross-modal learning frameworks to align sensor and text modalities, enabling zero-shot retrieval and captioning (Ndir et al., 2025; Zhang et al., 2025). Despite offering a promising pretraining paradigm, most of these methods fix the sensor encoder to a specific channel layout and sampling rate, so the pretrained model cannot be applied to data with a different input configuration without retraining (Zhang et al., 2025; Wang et al., 2025). More fundamentally, contrastive learning benefits from broad data diversity (Radford et al., 2021), yet these models pretrain on single-domain corpora, constraining the representation space they can learn.

To address these limitations, we propose SLIP, a unified sensor-language encoder that learns transferable representations via contrastive alignment across heterogeneous sensor configurations. SLIP departs from prior sensor-language models in three ways. First, we pretrain on a mixture corpus spanning multiple sensor domains, making the model sensor-agnostic: it assumes no fixed channel set or device type. Cross-domain negatives force the contrastive objective to align on semantic content rather than domain-specific signal characteristics. Second, this diversity means samples vary widely in sampling rate and length, and high-frequency modalities such as ECG at 500 Hz impose significant compute under a fixed patch size. To address this, we design *FlexMLP*, a patch embedder parameterized at a base patch size and resized on the fly per sample, making SLIP *length-adaptive*: a single checkpoint adjusts its patch size to each input's sampling rate and length, without resampling or retraining. Third, rather than training from scratch, we initialize SLIP from a pretrained decoder-only LM, partitioning its layers into a text encoder and a generative decoder. The resulting encoder-decoder supports both sensor-language understanding and open-vocabulary generation, unifying capabilities that prior work addresses separately. We summarize these differences in Table 1 and list our contributions below:

- **Unified and transferable sensor-language model:** We show that contrastive pretraining on a multi-domain mixture corpus yields sensor representations that transfer across heterogeneous configurations. With *FlexMLP* to handle varying sampling rates and lengths, a single checkpoint supports classification, retrieval, captioning, and QA across sensor domains without resampling or retraining.
- **Comprehensive multi-domain evaluation:** SLIP achieves 77.14% average linear-probing accuracy across four sensor domains, exceeding the best baseline by 3.56% and matching

supervised PatchTST (76.20%). On sensor QA, it achieves 68.87% average accuracy, outperforming all prior sensor-language models on every benchmark, demonstrating the transferability of SLIP across sensor configurations and temporal resolutions.

- **Curated sensor-language pretraining data:** To support sensor-language alignment at scale, we curate a dataset of 600K sensor-caption pairs spanning over 1 billion time points across diverse sensing domains including health, environment, Internet of Things (IoT), energy, and transportation, which we will release upon acceptance together with model weights and code to support further research.

2 Related Work

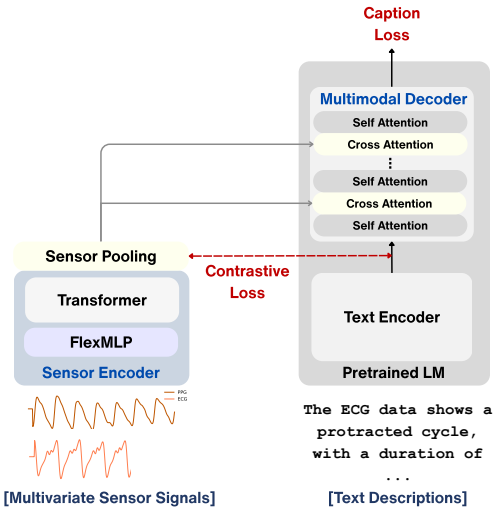
Time series and sensor foundation models. Sensor waveform signals from wearable devices are time series. Early efforts learn transferable representations via self-supervised pretraining with masked reconstruction (Dong et al., 2023; Nie et al., 2023) or contrastive learning (Zhang et al., 2022). More recently, general-purpose time series foundation models span domains including electricity, finance, and climate (Ansari et al., 2024; 2025; Woo et al., 2024; Goswami et al., 2024). While strong on forecasting, these models are optimized for generative objectives and often generalize poorly to classification, which dominates real-world sensor applications. Recent work also pretrains domain-specific sensor foundation models on large physiological and behavioral datasets (Abbaspourazad et al., 2023; Narayanswamy et al., 2024; Xu et al., 2025a; Qiu et al., 2025; Xu et al., 2025b). PPG-specific (Pillai et al., 2024; Saha et al., 2025) and ECG-specific (McKeen et al., 2025) models gain accuracy and robustness within their target domains. However, these models remain narrow and do not generalize across sensors and tasks.

LLMs for time series and sensor models. Recently, integrating language models into time series and sensor representation learning has emerged as a promising direction (Zhang et al., 2025; Langer et al., 2025; Pillai et al., 2025; Luo et al., 2025; Jin et al., 2023; Hu et al., 2025). Approaches such as Time-LLM or FSCA (Jin et al., 2023; Hu et al., 2025) reprogram time series inputs into the embedding space of pretrained large language models, enabling LLMs to capture high-level semantic structure in temporal data. In parallel, cross-modal approaches learn sensor representations through contrastive alignment with natural language. SensorLM (Zhang et al., 2025), for example, aligns wearable sensor streams from Fitbit devices with text using a hierarchical captioning pipeline and large-scale sensor-text pretraining, which improves zero-shot and few-shot recognition, cross-modal retrieval, and semantic grounding on human activity and health tasks. However, these approaches are largely restricted to specific sensor domains with fixed input lengths, which limits their generalization across diverse sensors and tasks. In contrast, we propose a unified language-informed sensor pretraining framework that supports heterogeneous sensor domains, enabling strong generalization across tasks ranging from classification to question answering.

3 SLIP Pretraining

Overview. We propose **SLIP** (Figure 2), a transferable sensor-language framework that learns aligned sensor-text representations from diverse sensor signals X_s and their paired text descriptions X_t. Specifically, SLIP is trained with a contrastive-captioner framework (Yu et al., 2022), grounding sensor representations in free-form language so that a single pretrained model can be queried, described, or classified through text. With *FlexMLP*, our patch embedder, SLIP adjusts its patch size to each input's length and sampling rate, so a single shared backbone encodes all inputs without resampling or retraining. This flexibility lets SLIP transfer easily across the heterogeneous healthcare sensing stack (from second-scale wearable streams to day-scale clinical signals) without per-dataset retraining or architectural changes. We describe each component in detail below.

Sensor Encoder. Behavioral and physiological representations are highly compressible since the patterns are repetitive. To capture this structure, SLIP uses a sensor encoder that compresses sensor inputs X_s into compact embeddings Z_s, a sequence of continuous vectors analogous to "tokens". Each channel is split into patches and embedded by *FlexMLP* (described below). The resulting per-channel patch tokens are concatenated into a single 1D sequence and processed by transformer blocks whose self-attention enables global cross-channel and long-range temporal interactions, yielding



Algorithm 1 Minimal FlexMLP pseudo-implementation.

```
class FlexMLP(nn.Module):
  def __call__(self, x, mask, time_index, base_patch=16):
    '''
    x, mask, time_index: (B, L)
    hidden_dim = 768
    mlp_dim = 3072
    '''

    x_p = patchify(x, patch_size)
    m_p = patchify(mask, patch_size)
    t_p = patchify(t, patch_size) # (B, num_patches, patch_size)
    z_p = concat([x_p, m_p, t_p], axis=-1) # (B, num_patches, patch_size*3)

    # Shared MLP weights trained at base_patch:
    w = self.param("w_mlp", (mlp_dim, base_patch*3))
    b = self.param("b_mlp", (mlp_dim,))
    w_res = self.param("w_res", (hidden_dim, base_patch*3))
    b_res = self.param("b_res", (hidden_dim,))

    # Flex trick: resize weights to match current patch size:
    w_p = resize(w, (mlp_dim, patch_size*3))
    w_res_p = resize(w_res, (hidden_dim, patch_size*3))
    h = linear(z_p, w_p, b)        # (B, num_patches, mlp_dim)
    r = linear(z_p, w_res_p, b_res) # (B, num_patches, hidden_dim)

    # Fixed output projection:
    h = silu(h)
    w_out = self.param("w_out", (hidden_dim, mlp_dim))
    b_out = self.param("b_out", (hidden_dim,))
    h = linear(h, w_out, b_out)    # (B, num_patches, hidden_dim)

    return dropout(h) + r
```

Figure 2: **SLIP architecture. Left:** SLIP aligns sensor signals and text with a contrastive captioner: a unified sensor encoder handles varying sensor configurations, and a pretrained LM serves as both text encoder and multimodal decoder. **Right:** Minimal FlexMLP pseudocode. MLP weights learned at `base_patch`$=16$ are resized to the runtime `patch_size`, so a single encoder handles variable-length inputs. Changes to existing code are highlighted.

the sensor embeddings Z_s. Within each block, we apply 2D RoPE (Su et al., 2023) to preserve the channel-time structure of the flattened sequence, so the model can distinguish each token by both its channel index and its temporal position. Following CoCa (Yu et al., 2022), an attention pooler with M learnable queries cross-attends over Z_s, producing a fixed-length output Z'_s. The first output token, z_{cls}, serves as the global sensor representation for the contrastive loss, and the remaining $M-1$ tokens are passed as prefix inputs to the multimodal decoder.

FlexMLP. Existing sensor and time series foundation models use a fixed patch size and resample inputs to a common rate (e.g., 65 Hz in Luo et al. (2025)), which either inflates token counts or discards fine-grained structure; we instead keep each signal at its native rate and adapt the patch size. FlexMLP is a patch-embedder whose single set of MLP weights is parameterized at a base patch size (i.e., 16) and resized on the fly to any runtime patch size (Beyer et al., 2023). With no additional parameters or computational overhead, FlexMLP processes long sensor inputs (e.g., EEG) at their native sampling rate by assigning them a larger patch size, bounding the number of tokens per input. Figure 2 (right) summarizes a general recipe to flexify an existing MLP patch-embedder (Ansari et al., 2025). We provide a detailed analysis and a case study in Appendix A.5 to demonstrate the practical value of FlexMLP.

Text Encoder and Multimodal Decoder. The *Text Encoder* $(X_t \mapsto Z_t)$ produces unimodal text representations Z_t for contrastive alignment, and the *Multimodal Decoder* $(\langle Z'_s, Z_t \rangle \mapsto \hat{X}_t)$ fuses these with the pooled sensor embeddings Z'_s to generate the target text $\hat{X}_t$ (Figure 2). Rather than train these branches from scratch (as in CoCa (Yu et al., 2022)), SLIP repurposes a single pretrained language model as the backbone of both: its lower layers serve as the text encoder, and its upper layers become the multimodal decoder by inserting a cross-attention sublayer into each block, so that text tokens can attend to Z'_s during autoregressive decoding. This lets SLIP inherit a strong language prior for both contrastive matching and free-form sensor description at zero extra pretraining cost.

Sensor Language Alignment. SLIP is optimized with two objectives jointly: (1) a contrastive loss (Radford et al., 2021) that aligns global sensor and text embeddings, scoring matched sensor-caption pairs above mismatched ones in the batch; and (2) a captioning loss that trains a multimodal decoder to autoregressively generate the caption, providing a denser supervision signal that captures finer-grained temporal structure, following CoCa (Yu et al., 2022).

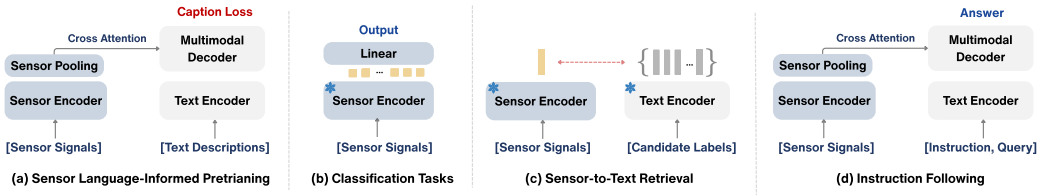


Figure 3: **Overview of the pretrained SLIP and its three downstream tasks:** (a) *pretraining* of SLIP via joint sensor-text contrastive matching and caption generation, (b) *classification* via linear probing on frozen sensor encoder features, (c) zero-shot *sensor-text retrieval* via cosine similarity between frozen sensor and text encoder embeddings, and (d) *instruction following* (sensor captioning and question answering) after supervised finetuning.

Pretraining Dataset Curation. Pretraining SLIP requires large-scale paired time series and text data, which is far less available than in vision-language settings. We start from community-released time series corpora (Liu et al., 2024; Luo et al., 2025) that provide diverse signals without aligned text, and we generate multi-level captions at statistical, structural, and semantic levels following the SensorLM recipe (Zhang et al., 2025). To increase pattern diversity, we augment with synthetic time series–text pairs from ChatTS (Xie et al., 2024). The resulting set contains over 600K samples and roughly one billion time points spanning energy, environment, health, IoT, nature, transportation, and web domains, with sampling rates from seconds to months and varied sequence lengths. To reduce template repetition, we prompt Qwen2-7B-IT (Yang et al., 2024) to generate three paraphrases per caption and randomly sample one during training. As in ChatTS, we sample multivariate and univariate examples in a 2:1 ratio. Corpus statistics are summarized in Table 6.

4 Experimental Setup

Implementation Details. SLIP features a sensor encoder and a text encoder-decoder, with 402M total parameters (189M trainable). The sensor encoder is a 120M-parameter Transformer (Vaswani et al., 2023; Nie et al., 2023) whose patch tokens are pooled via cross-attention with $M = 65$ learnable queries: a *classification* (CLS) token whose output serves as the global sensor embedding for computing the contrastive loss, and 64 caption queries that condition the multimodal decoder, following BLIP-2 (Li et al., 2023). The text branch is built on Gemma-3-270M (Team et al., 2025), with the first 12 layers serving as the unimodal text encoder and the final 6 layers turned into a multimodal decoder by inserting a cross-attention layer into each, so that text tokens can attend to sensor representations during autoregressive decoding. Full architectural choices and pretraining details are in Appendix A.3.

Downstream Tasks. As shown in Figure 3, we assess how SLIP transfers across three downstream tasks using standard evaluation protocols (Radford et al., 2021; Li et al., 2023):

- *Classification.* We train a linear classifier on top of the frozen sensor encoder features.
- *Retrieval.* Retrieval probes the cross-modal alignment between sensor and text representations in the shared embedding space, with no labeled adaptation at any stage (*i.e.*, zero-shot). We evaluate *sensor-to-text retrieval* only: each class label is converted into a templated sentence, and each sensor sample is assigned the label whose sentence embedding has the highest cosine similarity with its frozen sensor embedding.
- *Instruction following.* We finetune SLIP to perform both *sensor captioning*, where it generates a natural language description of the signals, and *question answering (QA)*, where it answers questions about the sensor data (e.g., identifying the activity given the accelerometer readings).

Metrics. We report top-1 accuracy for classification, retrieval, and QA, and evaluate sensor captioning with five text generation metrics: BLEU@4 (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), and ROUGE-L (Lin, 2004) measure the lexical n-gram overlap with the reference caption, while SBERTSimilarity (Reimers & Gurevych, 2019) and BERTScore (Zhang et al., 2020) capture semantic similarity through contextual embeddings. Higher is better for all metrics.

Table 2: **Evaluation of the SLIP on 11 downstream sensor tasks across 4 domains compared against multiple baselines with linear-probing.** We report the top-1 accuracy on the test set. Best results are in **bold** and second best are underlined.

			Activity Recognition			Clinical Diagnosis						Stress Prediction			Urban Sensing		
Category	Model	# Parameters	Average	WISDM	UCI-HAR	Average	Stroke	Diabetes	Hypertension	Sleep Stag.	Heart Cond.	Average	WESAD	Studentlife	Average	Obstacles	Beijing AQI
	PatchTST (Supervised)	*0.9M*	*85.08*	*85.66*	*84.49*	*76.52*	*90.91*	*82.96*	*48.48*	*84.29*	*75.94*	*62.20*	*75.78*	*48.62*	*80.51*	*84.91*	*76.11*
ML	Statistical ML	–	78.34	72.43	84.24	70.87	90.15	82.22	37.88	78.95	65.15	55.70	62.78	48.62	74.97	81.33	68.60
	SimMTM	0.5M	58.35	44.93	71.75	62.55	89.39	82.22	38.64	46.34	56.18	62.20	75.78	48.62	71.85	76.47	67.23
	TF-C	12M	59.60	55.39	63.81	63.66	90.91	82.96	39.39	46.97	58.06	49.58	56.95	42.20	47.98	49.87	46.08
TSFM	Sundial	128M	70.07	81.25	58.89	69.04	90.91	82.22	42.42	68.69	60.97	53.94	56.50	51.38	67.68	68.80	66.55
	Chronos	200M	81.07	80.30	81.84	70.85	81.06	77.78	46.97	76.68	71.76	57.50	66.37	48.62	77.02	75.19	**78.84**
	Chronos2	120M	77.74	75.02	80.45	71.89	90.91	77.04	47.73	82.48	61.27	60.40	73.09	47.71	65.67	81.84	49.49
	MOMENT	385M	71.35	69.99	72.71	71.98	82.88	78.22	45.91	80.06	**72.85**	60.01	71.75	48.26	73.76	85.68	61.84
SensorFM	NormWear	136M	74.98	70.77	79.19	73.00	90.91	82.22	40.15	**82.57**	69.15	63.99	80.27	47.71	79.06	83.38	74.74
	SensorLM	114M	82.70	76.57	**88.83**	71.32	85.45	75.11	44.24	80.74	71.05	64.08	79.55	48.61	79.63	83.22	76.04
	SLIP	120M	**84.31**	**82.36**	86.25	**75.79**	**91.67**	**84.44**	**50.00**	82.04	70.79	**68.55**	**82.96**	**54.13**	**81.96**	**86.45**	77.47

Datasets. For classification and retrieval tasks, we compile 11 datasets across 4 domains: *activity recognition* (WISDM (Weiss, 2019), UCI-HAR (Reyes-Ortiz et al., 2013)), *clinical diagnosis* (Stroke, Diabetes, Hypertension (Liang et al., 2018), Sleep Stage (Kemp et al., 2000), Heart Condition (Wagner et al., 2022)), *stress prediction* (WESAD (Schmidt et al., 2018), StudentLife (Wang et al., 2014)), and *urban sensing* (Obstacles (Souza, 2018), BeijingAQI (Chen, 2017)). All evaluation datasets are held out from our pretraining corpus. Urban sensing serves as an out-of-domain stress test, since our pretraining corpus contains limited in-domain data beyond healthcare. Table 7 summarizes the sensor modalities and label taxonomies for each dataset. For instruction following, we use five datasets (Langer et al., 2025): M4 for sensor captioning, and TSQA, HAR-CoT, Sleep-CoT, and ECG-QA-CoT for question answering. All four QA datasets follow the original free-form generation protocol, with TSQA probing basic sensor understanding and the three CoT datasets requiring chain-of-thought reasoning.

Baselines. Since each task in Figure 3 is supported by a different set of prior models, we select baselines per task. (i) For *classification*, we compare against three categories of representation learners: classical ML models (a random forest on catch22 features (Lubba et al., 2019), SimMTM (Dong et al., 2023), and TF-C (Zhang et al., 2022)), TSFMs (Sundial (Liu et al., 2025), Chronos (Ansari et al., 2024), Chronos-2 (Ansari et al., 2025), and MOMENT (Goswami et al., 2024)), and SensorFMs (NormWear (Luo et al., 2025) and SensorLM (Zhang et al., 2025)).[2] We additionally report a fully supervised PatchTST (Nie et al., 2023) as an upper-bound reference. (ii) For *retrieval*, we only compare against SensorFMs (NormWear and SensorLM), the only families that support sensor-text retrieval. (iii) For *instruction following* (captioning and QA), we compare against the soft-prompting (SP) and Flamingo variants of OpenTSLM (Langer et al., 2025) and SensorLM, all sharing the same Gemma-3-270M language backbone as SLIP for a controlled comparison.

5 Results & Discussion

5.1 Downstream Classification

Linear probing on frozen features is a standard protocol for evaluating the quality of pretrained representations (Radford et al., 2021). We adopt this protocol to compare representations learned with language supervision against those learned with predictive SSL objectives. Across 11 datasets spanning 4 domains (activity recognition, clinical diagnosis, stress prediction, and urban sensing), SLIP achieves the highest average accuracy in every domain (Table 2). The margin is largest on stress prediction (+4.56% over NormWear, the strongest predictive SSL baseline), where labels correspond to behavioral and physiological states (e.g., elevated heart rate, reduced movement, phone-usage patterns) that the pretraining captions describe explicitly. Language supervision therefore aligns the

[2]As SensorLM is not open-sourced, we reimplement it following Zhang et al. (2025) and pretrain it on the same corpus as SLIP.

Table 3: **Retrieval evaluation on 11 downstream tasks.** We compare SLIP with NormWear and SensorLM under the same sensor-text retrieval setting. All results report top-1 accuracy (R@1) on the test set. Best results are in **bold** and second best are underlined.

	Activity Classification			Clinical Diagnosis						Stress Prediction			Urban Sensing		
Model	Average	WISDM	UCI-HAR	Average	Stroke	Diabetes	Hypertension	Sleep Stag.	Heart Cond.	Average	WESAD	Studentlife	Average	Obstacles	Beijing AQI
NormWear	8.45	3.91	12.99	**52.52**	**89.39**	**82.22**	**36.36**	**39.09**	15.52	29.86	16.59	**43.12**	13.35	24.30	2.39
SensorLM	**17.86**	**9.66**	26.05	41.07	45.00	67.11	28.64	15.96	**48.64**	28.60	38.12	19.08	19.22	25.53	12.90
SLIP	17.54	7.45	**27.62**	49.11	86.40	74.07	34.85	34.83	15.39	**49.12**	**56.95**	41.28	**27.08**	**29.92**	**24.23**

embedding space directly with the factors that define these labels, making them linearly separable in the frozen features. Conversely, SLIP slightly underperforms predictive SSL baselines on Sleep Staging (-0.53% vs. NormWear) and Heart Condition (-2.06% vs. MOMENT). These tasks depend on fine-grained waveform morphology (e.g., sleep spindles, QRS-complex variations), which captions abstract away but signal-level objectives preserve. More broadly, both language-supervised models outperform all TSFMs pretrained with reconstruction objectives (77.14% average accuracy for SLIP and 73.58% for SensorLM). This result suggests that language supervision yields more transferable representations than raw-signal prediction. Because captions are invariant to sensor noise and other low-level nuisance factors, the language-supervised objective encourages the encoder to discard these factors and retain semantic structure. The learned representations are expressive enough that a linear probe on frozen SLIP features performs comparably to a fully supervised PatchTST trained end-to-end on each task. Compared with SensorLM, a controlled baseline trained on the same corpus with the same objectives, SLIP improves average accuracy by 3.56% and outperforms it on most datasets, demonstrating the effectiveness of its architectural design. A pairwise Wilcoxon test (Wilcoxon, 1945) confirms that SLIP differs significantly from every baseline at $p < 0.05$ (Appendix §A.4); accuracy over 5 random seeds and F1 results are provided in Tables 13 and 12.

5.2 Sensor-to-Text Retrieval

Although a joint sensor-language embedding space offers the flexibility to classify new sensor tasks from text descriptions alone, this setting is far harder than linear probing, since no classifier is fit to labeled data and accuracy depends entirely on the pretrained sensor-text alignment. NormWear addresses this by training an additional temporal fusion module that aligns its self-supervised sensor embeddings with text embeddings for zero-shot inference, and we therefore evaluate NormWear equipped with this module as the baseline in this setting. As Table 3 shows, SLIP achieves the highest average accuracy across all 11 datasets (39.36% vs. 33.26% for NormWear and 30.61% for SensorLM). The largest gain is on stress prediction, whose labels (e.g., stress, amusement) denote abstract affective states rather than directly observable signal patterns. SLIP inherits a text space that already encodes the meaning of such labels from its pretrained LM initialization, allowing contrastive pretraining to ground sensor patterns directly in these abstract concepts. On clinical diagnosis, NormWear outperforms SLIP slightly (52.52% vs. 49.11%) because its text encoder, ClinicalTinyLlama, is pretrained on clinical text and encodes concepts such as *hyperglycemia* and *autonomic neuropathy* that are rare in Gemma-3-270M's general-domain corpus. Overall, these results suggest that repurposing a pretrained LM, rather than learning text representations from sensor captions alone, carries broad language knowledge into the shared space and benefits sensor representation learning.

5.3 Instruction Following

Sensor Captioning. Unlike TSFMs, SLIP supports open-vocabulary generation, allowing it to answer arbitrary questions about sensor signals. We first evaluate this capability on sensor captioning, where the model describes the input signal in natural language (Figure 5). As shown in Table 4, SLIP outperforms both OpenTSLM variants across all metrics, from surface overlap (BLEU@4, ROUGE-L, METEOR) to semantic similarity (SBERTSimilarity, BERTScore). Moreover, we note

Table 4: **Sensor captioning and question answering results.** All models share the same Gemma-3-270M language backbone for a controlled comparison. Captioning is evaluated on the M4 test set with BLEU@4, METEOR, ROUGE-L, SBERTSim, and BERTScore. TSQA, HAR-CoT, Sleep-CoT, and ECG-QA-CoT share the same QA evaluation protocol, reporting top-1 accuracy. Higher is better for all metrics. Best results are in **bold** and second best are underlined.

Model	Sensor Captioning (M4)					TSQA	HAR-CoT	Sleep-CoT	ECG-QA-CoT
	BLEU@4	METEOR	ROUGE-L	SBERTSim	BERTScore	Accuracy	Accuracy	Accuracy	Accuracy
OpenTSLM-SP	0.0026	0.0456	0.0255	0.1551	0.7250	26.79	0.55	5.91	1.11
OpenTSLM-Flamingo	0.1141	0.3210	0.2894	0.7990	0.8858	78.12	63.43	68.49	35.50
SensorLM	0.2008	**0.4151**	0.3595	**0.9014**	0.9024	95.73	60.80	73.12	30.29
SLIP	**0.2046**	0.4027	**0.3658**	0.8977	**0.9077**	**99.75**	**64.35**	**74.19**	**37.18**

that SLIP performs on par with SensorLM, another language-aligned model, suggesting that language alignment provides a strong starting point for finetuning.

Sensor Question Answering. Beyond describing a signal, question answering requires the model to reason over it to reach a specific answer (Figures 6, 7, and 9). As shown in Table 4, in a controlled setting where all models have roughly the same number of parameters (120M sensor encoder, 270M language model), SLIP achieves the best accuracy on all four benchmarks, outperforming OpenTSLM and SensorLM. Specifically, the largest gain over OpenTSLM-Flamingo appears on TSQA (+21.63%), which mainly tests global signal patterns, suggesting that the contrastive objective captures high-level semantics while abstracting away low-level noise. Compared with SensorLM, SLIP performs significantly better on ECG-QA-CoT (+6.89%), which requires fine-grained waveform reasoning. As in retrieval, this result again suggests that initializing from a pretrained LM during pretraining helps the sensor encoder learn more fine-grained representations. Overall, the consistent margin across all four benchmarks suggests that language-aligned sensor representations provide more informative conditioning for the language model. We further compare against large TSLMs (i.e., ChatTS-8B and ITFormer-0.5B) in Table 17. Despite SLIP's much smaller language backbone, it outperforms both on all captioning metrics and most QA benchmarks.

5.4 Ablation Studies

In this section, we conduct an extensive ablation study on SLIP to understand how each architectural choice and pretraining objective affects the learned representations and downstream performance. All models follow the same evaluation protocol as in Section 4, and we report only the average performance for each task. Full results are provided in Appendix A.7.

Training Objectives. SLIP is jointly optimized with a contrastive loss and a captioning loss. To understand what each loss contributes, we train variants with only one of the two (Table 5(a)). Removing the captioning loss (*Contrastive-only*) barely affects retrieval (-2.46) but causes a large drop in QA (-15.81). Removing the contrastive loss (*Caption-only*) has the opposite effect: QA drops moderately (-6.02), while retrieval drops sharply (-14.24). These results suggest that the two objectives are complementary. The contrastive loss aligns a global sensor embedding with text, which supports zero-shot retrieval, whereas the captioning loss provides token-level supervision that supports generating answers but does not necessarily produce a well-structured joint embedding space. Appendix A.6 supports this with an alignment and uniformity analysis of the embedding space.

FlexMLP. Sequence lengths vary by orders of magnitude across datasets. FlexMLP therefore shares one set of patch-embedding weights across patch sizes, so a single pretrained encoder can tokenize each dataset at a suitable patch size without retraining. As Table 5 (b) shows, FlexMLP improves performance on every task, with the largest gain on retrieval ($+4.57$). Details of the patch-size heuristic are in Appendix A.3.2.

Cross-sensor Learning. Cross-channel interactions are typically modeled with either full self-attention over all channel-time tokens or memory-efficient group attention (Ansari et al., 2025), which restricts attention to within channel groups. In Table 5(c), the two perform comparably on perception and retrieval, while full attention is slightly better on QA ($+2.40$). We therefore keep full attention as the default and consider group attention a practical alternative under tight memory budgets.

Table 5: **Ablation study results**. The default setting adopted by SLIP is marked in blue. We calculate $\pm\ \Delta$ within each group of ablations in comparison with the default setting.

	Sensor Perception (Accuracy)		Retrieval (Accuracy)		QA (Accuracy)	
(a) Training Objectives						
SLIP pretraining	77.14		39.36		68.87	
Caption-only	74.30	(-2.84)	25.12	(-14.24)	62.85	(-6.02)
Contrastive-only	74.77	(-2.37)	36.90	(-2.46)	53.06	(-15.81)
(b) FlexMLP						
w/ FlexMLP	77.14		39.36		68.87	
w/o FlexMLP (*patch size* = 16)	74.16	(-2.98)	34.79	(-4.57)	66.88	(-1.99)
(c) Cross-Sensor Learning						
Self-Attention w/ 2D RoPE	77.14		39.36		68.87	
Group Attention	77.04	(-0.10)	39.21	(-0.15)	66.47	(-2.40)

	Sensor Perception (Accuracy)		Retrieval (Accuracy)		QA (Accuracy)	
(d) Sensor Encoder Parameter Size						
$\text{SLIP}_{\text{Base}}$ (120M)	77.14		39.36		68.87	
$\text{SLIP}_{\text{Small}}$ (40M)	74.84	(-2.30)	26.71	(-12.65)	59.93	(-8.94)
(e) Language Model						
Gemma-3-270M	77.14		39.36		68.87	
Llama3.2-1B	75.03	(-2.11)	36.53	(-2.83)	69.99	(+1.12)
(f) Partial finetuning of text-encoder						
Finetune 4-layer	77.14		39.36		68.87	
Freeze	73.56	(-3.58)	35.68	(-3.68)	60.55	(-8.32)

5.5 Further Analysis and Sensitivity Studies

Sensor and text encoder size. By default, SLIP pairs a 120M sensor encoder with a text encoder of comparable scale (Gemma-3-270M). We investigate whether a larger text encoder helps representation learning, as it encodes more real-world knowledge and may provide a better representation space for the sensor encoder to align with. We train SLIP with Llama-3.2-1B as the text encoder (Table 5 (e)), and surprisingly it underperforms the default Gemma-3-270M on sensor perception and retrieval, with only a marginal gain on QA ($+1.12$). This result suggests that it is more beneficial for the sensor and text encoders to have roughly the same number of parameters. To further validate this, we train a smaller sensor encoder (40M) with Gemma-3-270M and observe significant performance degradation on retrieval and QA, suggesting that the sensor features remain linearly separable but are poorly aligned with the text representation space. Together, these findings indicate that when one branch dominates in capacity, the contrastive objective provides weaker gradients to the smaller branch. We leave mitigations (*e.g.*, branch-specific learning rates or LoRA on the text branch) to future work.

Partial finetuning of text encoder. We explore whether freezing the text encoder allows the sensor encoder to learn a direct mapping into the text representation space, effectively translating sensor input into language. From Table 5 (f), we observe that doing so significantly degrades performance compared to our default SLIP, especially on QA (-8.32). As shown in Figure 4, sensor representations collapse toward a limited region of the feature space, leading to consistent degradation across downstream tasks. This result suggests that a frozen text encoder constrains the representation space too rigidly, forcing the sensor encoder to map all inputs into a narrow region rather than learning distributed representations. These results suggest that jointly finetuning both branches helps the two modalities co-adapt into a better-structured joint representation space.

6 Discussion and Conclusion

Limitations and future work. Two challenges remain open for sensor-language pretraining models. First, compared to vision-language models that rely on short captions, sensor-language models must condition on longer free-form descriptions to express signal semantics, increasing context length and computational cost. Scalability could be improved through selective decoding or adaptive context compression. Second, principled analyses of hallucination, output faithfulness, and temporal attribution remain largely absent for sensor-language generation. Developing such attribution and faithfulness methods remains an important direction for future work.

Conclusion and broader impact. This paper introduces SLIP, the first framework for learning transferable sensor representations from natural language supervision, trained with contrastive alignment and sensor-conditioned captioning in a single stage. By repurposing a pretrained decoder-only language model into a multimodal encoder-decoder via cross-attention, SLIP unifies perception, retrieval, and generation in one architecture. From a single checkpoint, SLIP achieves state-of-the-art performance across diverse sensor domains. It also adapts to complex sensor question answering with little finetuning. Unlike prior sensor encoders such as MOMENT, SLIP is, to our knowledge, the first that is both language-aligned and cross-modal across sensor types. By mapping heterogeneous sensor signals into a shared language-aligned space, SLIP can serve as a building block for future sensor-language models, and lower the barrier to downstream systems for health monitoring, activity understanding, accessibility, and human-centered AI. We hope this work motivates new directions for large sensor-language foundation models.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, for retrieval and discovery (e.g., finding related work), and for generating synthetic datasets. We have reviewed all AI-assisted work. Specifically, all AI-polished text was manually reviewed and revised by the authors to ensure accuracy and faithful representation of our contributions. Retrieved references were verified against original sources, and all synthetic datasets were validated for correctness before use in experiments. We did not use generative AI tools for research ideation or execution, to draft sections of the paper, or to prove mathematical claims. The remaining required disclosure categories are not applicable to this work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research has potential applications in healthcare and human-centered sensing, domains in which responsible data stewardship is paramount. Our study constitutes secondary analysis of de-identified, publicly available datasets and is therefore classified as non-human-subjects research under the Common Rule and analogous frameworks, requiring no additional Institutional Review Board (IRB) review at the authors' institution. All datasets used for pretraining and evaluation were either publicly released by their original authors under their respective licenses or made available in fully de-identified form from prior studies whose ethical approvals (IRB or equivalent) were secured by the original data providers, as documented in the cited source publications. No personally identifiable information was collected, accessed, or processed by the authors beyond what was provided in those public releases. Comprehensive details regarding dataset provenance, preprocessing, and intended use are provided in Appendix A.1. We further acknowledge that models trained on physiological and behavioral signals may carry risks of misuse, including unauthorized inference of sensitive attributes. We therefore encourage downstream practitioners to conduct domain-specific fairness, privacy, and safety assessments prior to deployment.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the full reproducibility of our work. All source code, preprocessed datasets, training and evaluation scripts, configuration files, and pretrained model weights used to produce the results reported in this paper will be publicly released upon acceptance under a permissive open-source license. The release will include: (i) complete implementation of the proposed method along with baseline re-implementations, (ii) the exact data splits, preprocessing pipelines, and random seeds required to reproduce every table and figure, (iii) trained model checkpoints together with inference scripts, and (iv) detailed documentation, environment specifications, and step-by-step instructions for replicating both training and evaluation. To further support reproducibility, all hyperparameters, hardware configurations, and training protocols are described in the main text and appendix. We believe these materials will allow the community to verify our findings and build upon our work.

REFERENCES

Salar Abbaspourazad, Oussama Elachqar, Andrew C Miller, Saba Emrani, Udhyakumar Nallasamy, and Ian Shapiro. Large-scale training of foundation models for wearable biosignals. *arXiv preprint arXiv:2312.05409*, 2023.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar 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. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael

Bohlke-Schneider. Chronos-2: From univariate to universal forecasting, 2025. URL https://arxiv.org/abs/2510.15821.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.

Richard B. Berry, Stuart F. Quan, Alexandre R. Abreu, Marietta L. Bibbs, Lourdes DelRosso, Susan M. Harding, Molly-Min Mao, David T. Plante, Mark R. Pressman, Matthew M. Troester, and Bradley V. Vaughn. *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. American Academy of Sleep Medicine, Darien, IL, version 2.6 edition, 2020. URL https://books.google.com/books?id=XT3dzQEACAAJ.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes, 2023. URL https://arxiv.org/abs/2212.08013.

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

Song Chen. Beijing Multi-Site Air Quality. UCI Machine Learning Repository, 2017. DOI: https://doi.org/10.24432/C5RK5G.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7: 1–30, December 2006. ISSN 1532-4435.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling, 2023. URL https://arxiv.org/abs/2302.00861.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llm capabilities in time series. *arXiv preprint arXiv:2501.03747*, 2025.

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

B. Kemp, A.H. Zwinderman, B. Tuk, H.A.C. Kamphuisen, and J.J.L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000. doi: 10.1109/10.867928.

Yubin Kim, Xuhai Xu, Daniel McDuff, Cynthia Breazeal, and Hae Won Park. Health-llm: Large language models for health prediction via wearable sensor data, 2024. URL https://arxiv.org/abs/2401.06866.

Patrick Langer, Thomas Kaar, Max Rosenblattl, Maxwell A. Xu, Winnie Chow, Martin Maritsch, Aradhana Verma, Brian Han, Daniel Seung Kim, Henry Chubb, Scott Ceresnak, Aydin Zahedivash, Alexander Tarlochan Singh Sandhu, Fatima Rodriguez, Daniel McDuff, Elgar Fleisch, Oliver Aalami, Filipe Barata, and Paul Schmiedmayer. Opentslm: Time-series language models for reasoning over multivariate medical text- and time-series data, 2025. URL https://arxiv.org/abs/2510.02410.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. Open spatio-temporal foundation models for traffic prediction. *ACM Transactions on Intelligent Systems and Technology*, October 2025. ISSN 2157-6912. doi: 10.1145/3773912. URL http://dx.doi.org/10.1145/3773912.

Yongbo Liang, Zhencheng Chen, Guiyong Liu, and Mohamed Elgendi. A new, short-recorded photoplethysmogram dataset for blood pressure monitoring in china. *Scientific Data*, 5:180020, 2018. doi: 10.1038/sdata.2018.20. URL https://doi.org/10.1038/sdata.2018.20.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models, 2024. URL https://arxiv.org/abs/2402.02368.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models, 2025. URL https://arxiv.org/abs/2502.00816.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. catch22: Canonical time-series characteristics, 2019. URL https://arxiv.org/abs/1901.10200.

Yunfei Luo, Yuliang Chen, Asif Salekin, and Tauhidur Rahman. Toward foundation model for multivariate wearable sensing of physiological signals, 2025. URL https://arxiv.org/abs/2412.09758.

Kaden McKeen, Sameer Masood, Augustin Toma, Barry Rubin, and Bo Wang. Ecg-fm: An open electrocardiogram foundation model. *JAMIA open*, 8(5):ooaf122, 2025.

Girish Narayanswamy, Xin Liu, Kumar Ayush, Yuzhe Yang, Xuhai Xu, Shun Liao, Jake Garrison, Shyam Tailor, Jake Sunshine, Yun Liu, et al. Scaling wearable foundation models. *arXiv preprint arXiv:2410.13638*, 2024.

Tidiane Camaret Ndir, Robin Tibor Schirrmeister, and Tonio Ball. Eeg-clip : Learning eeg representations from natural language descriptions, 2025. URL https://arxiv.org/abs/2503.16531.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Arvind Pillai, Dimitris Spathis, Fahim Kawsar, and Mohammad Malekzadeh. Papagei: Open foundation models for optical physiological signals. *arXiv preprint arXiv:2410.20542*, 2024.

Arvind Pillai, Dimitris Spathis, Subigya Nepal, Amanda C Collins, Daniel M Mackin, Michael V Heinz, Tess Z Griffin, Nicholas C Jacobson, and Andrew Campbell. Beyond prompting: Time2lang-bridging time-series foundation models and large language models for health sensing. In *Conference on Health, Inference, and Learning*, pp. 268–288. PMLR, 2025.

Minghui Qiu, Cekai Weng, Mingming Fan, and Kaishun Wu. Towards customizable foundation models for human activity recognition with wearable devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–29, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Jorge Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C54S4K.

Mithun Saha, Maxwell A Xu, Wanting Mao, Sameer Neupane, James M Rehg, and Santosh Kumar. Pulse-ppg: An open-source field-trained ppg foundation model for wearable applications across lab and field settings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–35, 2025.

Philip Schmidt, Attila Reiss, Robert Dürichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction (ICMI '18)*, pp. 400–408, New York, NY, USA, 2018. Association for Computing Machinery. doi: 10.1145/3242969.3242985.

Vinicius M. A. Souza. Asphalt pavement classification using smartphone accelerometer and complexity invariant distance. *Engineering Applications of Artificial Intelligence*, 74:198–211, 2018. ISSN 0952-1976. doi: 10.1016/j.engappai.2018.06.003. URL https://www.sciencedirect.com/science/article/pii/S0952197618301349.

Dimitris Spathis, Ignacio Perez-Pozuelo, Tomas I Gonzales, Yu Wu, Soren Brage, Nicholas Wareham, and Cecilia Mascolo. Longitudinal cardio-respiratory fitness prediction through wearables in free-living environments. *NPJ Digital Medicine*, 5(1):176, 2022.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton,

Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Rahul Thapa, Magnus Ruud Kjaer, Bryan He, Ian Covert, Hyatt Moore IV, Umaer Hanif, Gauri Ganjoo, M Brandon Westover, Poul Jennum, Andreas Brink-Kjaer, et al. A multimodal sleep foundation model for disease prediction. *Nature Medicine*, pp. 1–11, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Wojciech Samek, and Tobias Schaeffter. PTB-XL, a large publicly available electrocardiography dataset. *PhysioNet*, November 2022. doi: 10.13026/kfzx-aw45. URL https://doi.org/10.13026/kfzx-aw45. Version 1.0.3.

Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben Zeev, and Andrew T. Campbell. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, New York, NY, USA, 2014. Association for Computing Machinery. doi: 10.1145/2632048.2632054.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2022. URL https://arxiv.org/abs/2005.10242.

Yilin Wang, Peixuan Lei, Jie Song, Yuzhe Hao, Tao Chen, Yuxuan Zhang, Lei Jia, Yuanxiang Li, and Zhongyu Wei. Itformer: Bridging time series and natural language for multi-modal qa with large-scale multitask dataset, 2025. URL https://arxiv.org/abs/2506.20093.

Gary Weiss. WISDM Smartphone and Smartwatch Activity and Biometrics Dataset . UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C5HK59.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024. URL https://arxiv.org/abs/2402.02592.

Zhe Xie, Zeyan Li, Xiao He, Longlong Xu, Xidao Wen, Tieying Zhang, Jianjun Chen, Rui Shi, and Dan Pei. Chatts: Aligning time series with llms via synthetic data for enhanced understanding and reasoning. *arXiv preprint arXiv:2412.03104*, 2024.

Maxwell A Xu, Girish Narayanswamy, Kumar Ayush, Dimitris Spathis, Shun Liao, Shyam A Tailor, Ahmed Metwally, A Ali Heydari, Yuwei Zhang, Jake Garrison, et al. Lsm-2: Learning from incomplete wearable sensor data. *arXiv preprint arXiv:2506.05321*, 2025a.

Wenxuan Xu, Arvind Pillai, Subigya Nepal, Amanda C Collins, Daniel M Mackin, Michael V Heinz, Tess Z Griffin, Nicholas C Jacobson, and Andrew Campbell. Lens: Llm-enabled narrative synthesis for mental health by aligning multimodal sensing with language models, 2025b. URL https://arxiv.org/abs/2512.23025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. URL https://arxiv.org/abs/2205.01917.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency, 2022. URL https://arxiv.org/abs/2206.08496.

Yuwei Zhang, Kumar Ayush, Siyuan Qiao, A. Ali Heydari, Girish Narayanswamy, Maxwell A. Xu, Ahmed A. Metwally, Shawn Xu, Jake Garrison, Xuhai Xu, Tim Althoff, Yun Liu, Pushmeet Kohli, Jiening Zhan, Mark Malhotra, Shwetak Patel, Cecilia Mascolo, Xin Liu, Daniel McDuff, and Yuzhe Yang. Sensorlm: Learning the language of wearable sensors, 2025. URL https://arxiv.org/abs/2506.09108.

A Appendix

A.1 Datasets

We thank the authors of prior work who collected and open-sourced the datasets used in this work, and we summarize them again below for the convenience of future researchers. We provide the details of the non-overlapping pretraining and downstream datasets below.

A.1.1 Pretraining Dataset

Inspired by SensorLM (Zhang et al., 2025), we automatically generate hierarchical captions (i.e., statistical, structural, semantic) of each multivariate sensor signal. The domain distribution of this sensor-paired dataset is shown in Table 6. We will open-source this large dataset to support future research on sensor-language models, and we will also release the caption generation pipeline for creating large-scale sensor-language datasets.

Table 6: Category distribution.

	Health	Web	Synthetic	Nature	Energy	IoT	Environment	Transport	Total
# of Samples	285054	145063	105085	69518	5846	5508	2368	66	618508
Percent (%)	46.09	23.45	16.99	11.24	0.95	0.89	0.38	0.01	100.00
Sources	(Liu et al., 2024) (Luo et al., 2025) (Chan et al., 2024)	(Liu et al., 2024)	(Xie et al., 2024)	(Liu et al., 2024)	(Liu et al., 2024)	(Liu et al., 2024)	(Liu et al., 2024)	(Liu et al., 2024)	

A.1.2 Downstream Dataset

Table 7: Evaluation dataset details.

Dataset	Sensor	# Samples (Train / Test)	Freq.	# Cls	Label Names
WISDM (Weiss, 2019)	Accelerometer X, Y, Z	22396 / 5600	30Hz	18	Catch, Chips, Clap, Dribble, Drink, Fold, Jog, Kick, Pasta, Sandwich, Sit, Soup, Stair, Stand, Teeth, Type, Walk, Write
UCI-HAR (Reyes-Ortiz et al., 2013)	Accelerometer X, Y, Z Gyroscope X, Y, Z	1847 / 793	50Hz	7	Lay, Sit, Stand, Walk, Walking Upstairs, Walking Downstairs, Transition
PPG-CVA (Stroke) (Liang et al., 2018)	PPG	525 / 132	65Hz	2	Normal, Stroke
PPG-DM (Diabetes) (Liang et al., 2018)	PPG	522 / 135	65Hz	2	Normal, Diabetes
PPG-HTN (Hypertension) (Liang et al., 2018)	PPG	525 / 132	65Hz	4	Normal, Pre-hypertension, Stage-1, Stage-2

Dataset	Sensor	# Samples (Train / Test)	Freq.	# Cls	Label Names
Sleep Stage (Sleep stage) (Kemp et al., 2000)	EEG-Fpz-Cz, EEG-Pz-Oz	33599 / 8709	100Hz	5	Light spindle, Light, Deep, REM, Wake
PTB-XL (Heart cond.) (Wagner et al., 2022)	12-lead ECG	11320 / 1650	100Hz	5	Normal ECG, Myocardial Infarction, ST/T Change, Conduction Disturbance, Hypertrophy
WESAD (Schmidt et al., 2018)	Chest Acc. X, Y, Z Chest ECG, EMG, EDA, Temp, Resp, Wrist Acc. X, Y, Z, Wrist BVP, EDA	882 / 223	700Hz	3	Neutral, Amusement, Stress
StudentLife (Wang et al., 2014)	Activity, Audio, Conversation, Phone Charge, Phone Lock, Time to deadline, Day of the week, Exam period, Sleep rating, Sleep duration	1074 / 109	Minute	3	Normal, Medium Stress, High Stress
AsphaltObstacles (Souza, 2018)	Acceleration magnitude	390 / 391	100Hz	4	Raised crosswalk, Raised markers, Speed bump, Vertical patch
Beijing AQI (Chen, 2017)	dew-point temperature, windspeed, PM25, PM10, NO2, SO2, CO	1168 / 293	Hour	5	Good, Moderate, Unhealthy for sensitive groups, Unhealthy, Hazardous

A.2 Pretraining Dataset Scaling

Table 8 illustrates the scaling behavior of SLIP across different data regimes. Consistent with the scaling laws observed in the original CLIP work, our results demonstrate that retrieval performance benefits significantly more from large-scale data than classification. While classification accuracy remains relatively stable—dropping only 2.88% when scaling down from 1.7B to 0.3B samples—retrieval performance collapses by 17.72%. This follows the intuition from Radford et al. (2021) that while coarse semantic categories can be learned from limited samples, the construction of a high-fidelity, shared latent space requires the "density" of a massive dataset. Large-scale pretraining is essential for the model to learn the fine-grained nuances necessary to distinguish between similar sensor-text pairs in a global retrieval space, suggesting SLIP's greater potential as datasets continue to grow.

Table 8: **Effect of Pretraining Dataset Size**

Model	**Cls. Avg**	**Δ**	**Retrieval**	**Δ**	**QA**	**Δ**
Ours (1.7 B)	77.14		39.36		68.87	
0.8 B	75.23	(-1.91)	27.09	(-12.27)	60.18	(-8.69)
0.3 B	74.26	(-2.88)	21.64	(-17.72)	62.95	(-5.92)

A.3 Implementation Details

A.3.1 Architecture

Sensor encoder. The sensor encoder is a standard Transformer backbone with 120M parameters and hidden dimension 768 (Vaswani et al., 2023; Nie et al., 2023). Following Liu et al. (2025), we adopt Pre-LN and FlashAttention for training stability and efficiency, and use 2D RoPE (Su et al., 2023) to preserve the channel-time structure across the flattened token sequence. Instead of the conventional linear projection from input signal patches to hidden dimension, we use FlexMLP with a base patch size of 16 to support variable patch sizes for different sampling rates (see Appendix A.3.2 for the patch size heuristic). The resulting tokens are pooled via cross-attention with $M = 65$ learnable queries: one CLS query for contrastive matching, and 64 caption queries that condition the multimodal decoder, corresponding to an $8\times$ temporal compression as in BLIP-2 (Li et al., 2023). During pretraining, we cap each example at 6 channels and 384 sensor patches to keep training efficient and enable larger contrastive batch sizes.

Text encoder-decoder. The text encoder is initialized from the first 12 layers of Gemma-3-270M (Team et al., 2025), placing its parameter budget in the same range as text encoders commonly used in CoCa, while the multimodal decoder is initialized from the final 6 layers of the same model. To condition text generation on sensor representations, we insert a cross-attention layer into each of the final six unimodal layers, transforming them into a multimodal decoder in which text tokens can attend to sensor encoder outputs during autoregressive decoding. For efficiency and training stability, we unfreeze only the last 4 layers of the unimodal text encoder, yielding 22M trainable parameters in the encoder. The effect of freezing the entire text encoder is analyzed in Section 5.4. We reuse the Gemma-3-270M tokenizer and token embedding, and cap the input at 512 query tokens, padding shorter queries with [PAD]. After adding cross-attention, the decoder is trained with full parameters (43M trainable). Overall, this branch closely follows the Gemma-3-270M architecture and introduces only 10M additional parameters via cross-attention.

A.3.2 Patch Size Heuristic

Our pretraining corpus mixes sampling rates from monthly to 700Hz and sequence lengths that differ by orders of magnitude (Table 6). A single patch size would turn this diversity into padding: short low-rate series would be padded out to the length of long high-rate ones, and most of the compute in a batch would be spent on padding patches. The patch size heuristic is what lets us train on the whole corpus jointly. It assigns a larger patch to longer or lower-rate series, so the token count per sample stays roughly constant across datasets and every token carries real signal. We define the heuristic by frequency, following Woo et al. (2024), and widen its range of window sizes to cover our corpus:

- **Daily:** 16, 32
- **Hourly:** 6, 8, 16, 24, 32, 64
- **Minute-level:** 16, 24, 128
- **Second-level:** 4, 6, 8, 12, 16, 20, 25, 32, 64, 128

Because the heuristic assigns different patch sizes across frequencies, a single set of patch embedding weights is shared across patch sizes and resized by interpolation at evaluation time. The patch size is thus a choice made after pretraining: a user can trade token count for temporal granularity according to the downstream task, without retraining the encoder. We analyze this behavior in Appendix A.5.

A.3.3 Pretraining Implementation Details

We pretrain for 40 epochs with batch size 72 on 4 NVIDIA H200 GPUs using AdamW (Loshchilov & Hutter, 2019) (weight decay = 0.05). The learning rate is warmed up to $2e^{-4}$ over 80k iterations and cosine-decayed to $1e^{-7}$. Sensor augmentations are disabled during pretraining to preserve sensor-text alignment, but standard augmentations (jittering, scaling, time flipping) are applied during supervised finetuning (Zhang et al., 2025). Additional implementation details are in the codebase.

A.3.4 Downstream Evaluation Details

Following the standard transfer learning evaluation protocol from prior work (He et al., 2021), we did not perform hyperparameter search (e.g., learning rate or weight decay). All models were frozen, and only a randomly initialized linear classifier was trained for 50 epochs with 5 warmup epochs using the AdamW optimizer, a base learning rate of 0.01, and a weight decay of 0.05. Supervised finetuning hyperparameter details are shown in Table 9.

Table 9: **Hyperparameters used in the supervised finetuning experiment.**

Hyperparameter	TSQA	HAR-CoT	Sleep-CoT	ECG-QA-CoT	M4 Caption
Optimizer	AdamW (Loshchilov & Hutter, 2019)				
Gradient clip	1.0				
LR decay schedule	Cosine Schedule Decaying to $1e^{-7}$				
Train Epoch	4	4	10	4	4
Train batch size	64	64	32	32	64
Warm up epochs	1	1	1	1	1
Weight decay rate	0.05	0.05	0.05	0.05	0.05

A.4 Statistical Comparisons

Following Demšar (2006), we first run a Friedman omnibus test on the linear-probing accuracies of all 10 models across the 11 datasets in Table 2, excluding the supervised PatchTST reference. The test gives $\chi_F^2 = 32.7$ with $p = 1.5 \times 10^{-4}$, firmly rejecting the null that all models perform identically. We then compare SLIP against each baseline using a two-sided paired Wilcoxon signed-rank test over the 11 datasets, and report the results in Table 10. SLIP attains the best average rank (1.64) and improves over every baseline at $p \leq 0.01$, which remains significant after Holm correction for the 9 comparisons (adjusted $p \leq 0.015$).

For the zero-shot setting we apply the same test against both retrieval baselines. SLIP outperforms SensorLM on 9 of the 11 datasets with a median margin of 6.96% ($W = 12$, $p = 0.067$), while it is statistically indistinguishable from NormWear ($W = 24$, $p = 0.46$, winning on 5 of 11 datasets). The latter is consistent with NormWear's stronger alignment on the diagnosis tasks. We do not report statistical tests for the supervised finetuning experiment because there are fewer than five datasets, following the guidance in Demšar (2006).

Table 10: **Wilcoxon signed-rank tests of SLIP against each baseline.** Average ranks are over the 11 datasets, lower is better. $^{*}p < 0.1$, $^{**}p < 0.05$, n.s. = not significant.

Eval	Model	Avg. rank	p	sig
LP	**SLIP (ours)**	**1.64**	–	–
	SensorLM	4.64	0.0098	**
	NormWear	4.68	0.0020	**
	Chronos	5.18	0.0049	**
	Chronos2	5.55	0.0020	**
	MOMENT	5.82	0.0049	**
	Stat Feat	5.86	0.0010	**
	Sundial	6.36	0.0010	**
	SimMTM	7.23	0.0010	**
	TFC	8.05	0.0010	**
ZS	**SLIP (ours)**	**1.73**	–	–
	NormWear	2.00	0.4648	n.s.
	SensorLM	2.27	0.0674	*

A.5 FLEXMLP

FlexMLP equalizes sequence length across sampling rates by varying the patch size. The same equalization can be reached by resampling all data to one rate instead, as in NormWear (65Hz). Resampling shortens the sequence but also discards the high-frequency content of a signal before the model sees it, and that content often carries the label. Table 11a shows this on Sleep Stage: sleep spindles are removed at 25Hz, and every rhythm above the slow waves is removed at 5Hz. Enlarging the patch shortens the sequence by the same factor while keeping the signal.

Table 11b compares the two at equal tokens per channel, row by row. FlexMLP enlarges the patch and keeps the recorded 100Hz signal; downsampling fixes the patch at 16 and lowers the rate. The last column is the gain of FlexMLP. Accuracy falls along both columns, since a shorter sequence carries less information either way. The gap between them tracks the bands in Table 11a. The two stay on par at 94 and 47 tokens. They separate at 10, where downsampling has removed the bands that carry the label, and a patch of 300 gives 72.21 against 66.17. Enlarging the patch therefore relieves the loss rather than removing it: accuracy still falls 10.82% between 188 and 10 tokens, and what it avoids is giving up the bandwidth on top.

One set of embedding weights serves every patch size, resized to whichever one the heuristic picks. The FlexMLP column measures what that resizing costs. Every row sees the same recorded 100Hz signal, so moving from a patch of 16 to 32 changes only the resizing and halves the sequence, at a cost of 0.99%. The resized kernel keeps the spectral structure that carries the label, even though its frequency response has changed. The larger drops further down the column come from the token budget itself.

Table 11: **Enlarging the patch versus downsampling at equal sequence length.** (a) EEG rhythms retained at each sampling rate, on a 30s Sleep Stage window, with bands from the AASM manual (Berry et al., 2020). (b) Sleep Stage linear probing accuracy at equal tokens per channel. Reference settings are marked in blue.

(a)

	100Hz (native)	50Hz	25Hz	5Hz
Slow wave activity (0.5–2Hz)	kept	kept	kept	kept
Theta (4–8Hz)	kept	kept	kept	removed
Alpha (8–13Hz)	kept	kept	partial	removed
Sleep spindles (11–16Hz)	kept	kept	removed	removed
Beta (13–30Hz)	kept	partial	removed	removed

(b)

	FlexMLP		Downsampling		
Token budget	Patch size	Acc.	Rate	Acc.	Gain
188	16	83.03	100Hz	83.03	–
94	32	82.04	50Hz	82.62	(-0.58)
47	64	79.01	25Hz	78.96	(+0.05)
10	300	**72.21**	5Hz	66.17	(+6.04)

A.6 Uniformity and Alignment

Setup. To better understand the geometric properties of the embedding space, we follow Wang & Isola (2022). The key idea is to view contrastive learning as balancing two geometric effects in the embedding space. Alignment measures whether a model pulls together embeddings of positive pairs that should represent the same underlying content, while uniformity measures whether embeddings of different samples remain spread out on the unit hypersphere instead of collapsing. In their original image setting, a positive pair is two augmented views of the same image, and negatives are views from other images in the batch. Good representations therefore require low alignment loss, meaning the two views of the same image land close, while also maintaining good uniformity, meaning the whole set of image embeddings does not crowd into a small region.

In sensor-language contrastive learning, the same definitions apply after replacing the notion of a positive pair. Here, the positive pair is a matched sensor segment x and its paired text description y, and negatives are mismatched sensor-text pairs (x_i, y_j) with $i \neq j$. Alignment then becomes a direct measure of sensor-to-language coupling: if the representation is truly shared, embeddings of matched sensor and text should be close. Namely,

$$\mathcal{L}_{\text{align}} \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}^+}[\|f(x) - f(y)\|_2^\alpha], \quad \alpha > 0.$$

Uniformity becomes a check against collapse within each modality: sensor embeddings should remain spread across the sphere so different signals are distinguishable, and text embeddings should remain spread so different descriptions are distinguishable. The uniformity loss is defined as the logarithm of the average pairwise Gaussian potential:

$$\begin{aligned}\mathcal{L}_{\text{unif}}(f;t) &\triangleq \log \mathbb{E}_{x,y \overset{\text{iid}}{\sim} \mathcal{D}}\left[G_t(x,y)\right] \\ &= \log \mathbb{E}_{x,y \overset{\text{iid}}{\sim} \mathcal{D}}\left[e^{-t\|f(x)-f(y)\|_2^2}\right], \quad t > 0.\end{aligned}$$

This is especially useful for sensor data because many windows can be similar and the model can otherwise reduce loss by relying on the stronger text branch, so tracking sensor uniformity alongside cross-modal alignment helps distinguish true cross-modal alignment from a degenerate solution. Altogether, we measure sensor-language alignment, sensor uniformity, and text uniformity with implementation in Alg. 1.

Result. Figure 4 shows that the caption-only model suffers from poor sensor-text alignment and degraded sensor and text uniformity, consistent with representation collapse. Notably, when freezing the text encoder, sensor features spread more slowly, plateau earlier, and exhibit weaker sensor-text alignment.

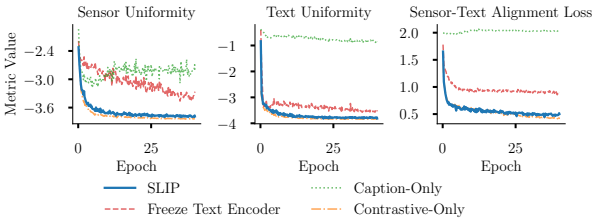


Figure 4: **Sensor-Language representation geometry analysis. Sensor Uniformity (left)** and **Text Uniformity (middle)** quantify embedding dispersion on the unit hypersphere; **Sensor-Text Alignment (right)** measures mean distance between paired sensor and text embeddings. Lower is better.

Algorithm 1 Implementation of alignment and uniformity.

```
import torch
import torch.nn.functional as F

# bsz: batch size, d: latent dim
# x: (bsz, d) sensor emb
# y: (bsz, d) text emb

x = F.normalize(x, dim=1)
y = F.normalize(y, dim=1)

def lalign(x, y, alpha=2):
    return (x - y).norm(dim=1)\
        .pow(alpha).mean()

def lunif(x, t=2):
    sq = torch.pdist(x, p=2).pow(2)
    return sq.mul(-t).exp().mean().
     log()

align_loss   = lalign(x, y)
txt_unif     = lunif(y)
sensor_unif  = lunif(x)
```

A.7 SUPPLEMENTARY RESULTS

Table 12: **Additional F1 scores.** Classification (LP) and retrieval (ZS), mirroring Table 2.

Eval	Model	WISDM	UCIHAR	Stroke	Diabetes	Hypertension	Sleep Stage	Heart Cond.	WESAD	StudentLife	Obstacles	Beijing AQI	Average
LP	Stat Feat	72.03	84.18	47.41	45.12	26.85	66.41	39.12	47.81	**42.46**	80.98	36.86	53.57
	SimMTM	42.93	69.13	47.20	31.35	**45.12**	28.68	17.17	54.56	35.86	75.85	37.32	44.11
	TFC	54.60	64.20	47.62	45.12	23.86	33.08	24.50	23.60	25.11	47.40	23.44	37.50
	NormWear	70.60	76.88	52.30	44.90	32.99	72.25	47.52	63.02	32.24	82.82	48.58	56.74
	Chronos2	74.43	81.41	47.41	43.28	34.96	**73.83**	35.76	57.26	26.35	78.83	25.28	52.62
	Chronos	80.06	81.73	47.35	**57.70**	35.54	66.07	48.89	51.72	28.17	74.61	**61.17**	57.47
	Sundial Base	80.90	41.64	47.62	45.12	13.54	51.18	30.53	36.85	38.41	67.68	36.15	44.51
	ChatTS	75.36	57.31	47.62	45.12	13.93	69.70	42.36	57.31	36.19	76.12	29.85	50.08
	SensorLM	76.17	84.03	50.15	52.06	37.12	70.90	**50.06**	69.14	42.31	82.88	52.24	60.64
	SLIP	**81.87**	**84.58**	**54.74**	51.71	39.59	70.98	48.67	**71.84**	40.62	**84.32**	45.32	**61.17**
ZS	NormWear	0.48	3.89	**47.20**	45.12	13.48	11.24	5.37	9.49	20.09	14.00	1.83	15.65
	SensorLM	4.26	12.87	38.42	46.18	17.60	12.07	**18.01**	36.00	14.04	10.73	12.19	20.22
	SLIP	**5.47**	**13.80**	46.34	**45.19**	**17.15**	**29.22**	5.36	**46.02**	**24.97**	**15.33**	**13.81**	**23.88**

Table 13: **Additional supplementary results.** Classification (LP) and retrieval (ZS), mean ± std over 5 random seeds.

Eval	Model	WISDM	UCIHAR	Stroke	Diabetes	Hypertension	Sleep Stage	Heart Cond.	WESAD	StudentLife	Obstacles	Beijing AQI
LP	Stat Feat	77.14 ± 0.25	85.62 ± 0.38	90.15 ± 0.00	82.22 ± 0.00	37.73 ± 1.30	77.33 ± 0.02	60.99 ± 0.33	68.52 ± 0.77	48.26 ± 1.37	81.69 ± 0.80	68.81 ± 0.51
	SimMTM	33.76 ± 0.08	70.52 ± 0.24	89.39 ± 0.00	82.22 ± 0.00	37.88 ± 1.07	41.76 ± 0.07	55.90 ± 0.08	62.51 ± 0.46	50.64 ± 0.69	36.37 ± 0.98	71.26 ± 0.26
	TFC	44.11 ± 0.11	59.95 ± 0.13	89.85 ± 0.61	80.74 ± 0.47	40.00 ± 0.57	44.88 ± 0.05	58.11 ± 0.25	55.43 ± 0.46	43.67 ± 1.70	33.66 ± 0.62	45.60 ± 0.46
	NormWear	70.82 ± 0.03	79.37 ± 0.32	90.45 ± 0.61	82.37 ± 0.30	40.45 ± 1.13	82.92 ± 0.18	69.53 ± 0.07	80.27 ± 0.49	49.17 ± 1.24	83.38 ± 0.32	74.40 ± 0.38
	Chronos2	75.18 ± 0.07	75.38 ± 0.51	87.12 ± 0.96	73.33 ± 3.18	38.79 ± 0.88	**83.74** ± 0.14	61.20 ± 0.19	67.26 ± 0.75	47.89 ± 0.69	84.14 ± 0.23	51.74 ± 0.63
	Chronos	81.19 ± 0.13	84.04 ± 0.30	88.48 ± 2.00	80.59 ± 1.36	45.30 ± 0.88	76.69 ± 0.14	**73.27** ± 0.18	66.46 ± 0.18	49.54 ± 1.00	75.45 ± 0.28	**80.48** ± 0.14
	Sundial Base	41.39 ± 0.12	55.84 ± 0.29	77.27 ± 0.83	72.00 ± 5.03	37.73 ± 3.16	70.19 ± 0.14	61.26 ± 0.12	59.64 ± 0.40	47.53 ± 2.20	68.04 ± 0.23	67.85 ± 0.26
	MOMENT	69.99 ± 0.08	72.71 ± 0.41	82.88 ± 6.58	78.22 ± 3.30	45.91 ± 1.03	80.06 ± 0.03	72.85 ± 0.07	71.75 ± 0.28	48.26 ± 1.10	**85.68** ± 0.16	61.84 ± 0.79
	ChatTS	75.66 ± 0.00	56.24 ± 0.00	90.91 ± 0.00	82.22 ± 0.00	44.70 ± 0.00	80.65 ± 0.08	67.93 ± 0.21	77.04 ± 0.18	50.28 ± 0.90	77.75 ± 0.65	61.23 ± 1.84
	SensorLM	76.57 ± 0.18	**88.83** ± 0.29	85.45 ± 1.11	75.11 ± 0.76	44.24 ± 0.61	80.74 ± 0.05	71.05 ± 0.08	79.55 ± 0.46	48.61 ± 0.90	83.22 ± 0.31	76.04 ± 0.33
	SLIP	**82.28** ± 0.10	85.67 ± 0.26	**91.36** ± 0.67	**83.11** ± 0.55	**47.58** ± 0.57	82.62 ± 0.09	71.59 ± 0.19	**81.79** ± 0.61	**54.31** ± 1.58	85.32 ± 0.35	76.66 ± 0.27
ZS	NormWear	3.91 ± 0.66	12.98 ± 2.15	89.40 ± 3.74	**82.22** ± 7.18	**36.38** ± 8.38	**39.09** ± 0.73	15.52 ± 1.26	16.58 ± 6.43	**43.12** ± 13.02	24.30 ± 2.03	2.40 ± 2.34
	SensorLM	**9.66** ± 0.26	26.05 ± 0.54	45.00 ± 3.57	67.11 ± 2.46	28.64 ± 2.64	15.96 ± 0.25	**48.64** ± 0.32	38.12 ± 2.02	19.08 ± 1.35	25.53 ± 0.38	12.90 ± 1.64
	SLIP	7.45 ± 0.14	**27.26** ± 0.46	**90.91** ± 0.00	75.56 ± 2.65	35.76 ± 0.30	34.58 ± 0.17	16.08 ± 0.19	**40.45** ± 0.87	42.02 ± 0.69	**30.08** ± 0.13	**22.25** ± 1.67

Table 14: **Ablation Studies Linear Probing Performance.** Reported Top-1 accuracy on 11 datasets.

Model	WISDM	UCIHAR	Stroke	Diabetes	Hypertension	Sleep Stage	Heart Cond.	WESAD	StudentLife	Obstacles	Beijing AQI	All Avg
SLIP	**82.36**	86.25	91.67	84.44	**50.00**	82.04	70.79	82.96	54.13	**86.45**	77.47	**77.14**
Contrastive-Only	79.14	85.37	90.91	84.44	40.91	81.10	68.12	82.96	50.46	84.65	74.40	74.77
Caption-Only	82.11	84.36	90.91	83.70	37.12	81.96	69.70	76.68	46.79	85.42	78.50	74.30
Random Paired	38.16	65.45	90.91	82.96	41.67	59.62	55.39	52.47	52.47	74.93	69.62	62.15
w/o *FlexMLP* ($ps = 16$)	76.80	**89.16**	87.88	82.22	47.73	82.55	70.73	72.20	46.79	84.91	74.74	74.16
w/ Group Attention	79.45	84.36	**92.42**	**85.19**	48.48	81.67	**72.12**	83.41	**56.88**	85.94	77.47	77.04
Freeze Text Encoder	76.75	85.12	83.33	79.26	40.91	79.21	66.55	**87.44**	54.13	82.35	74.06	73.56
Llama3.2-1B Backbone	82.16	86.63	88.79	81.63	38.03	**82.75**	71.02	80.36	51.19	85.52	77.27	75.03
$\text{SLIP}_{\text{Small}}$ (40M)	80.54	81.84	90.91	83.70	42.42	80.74	68.67	81.17	49.54	84.91	**78.84**	74.84

Table 15: **Ablation Studies Retrieval Performance.** Top-1 accuracy on 11 datasets.

Model	WISDM	UCIHAR	Stroke	Diabetes	Hypertension	Sleep Stage	Heart Cond.	WESAD	StudentLife	Obstacles	Beijing AQI	All Avg
SLIP	7.45	**27.62**	86.40	74.07	34.85	34.83	15.39	**56.95**	41.28	29.92	24.23	**39.36**
Contrastive-Only	9.50	19.67	79.55	80.74	24.24	39.05	10.97	52.47	40.37	30.18	19.11	36.90
Caption-Only	5.82	6.56	78.03	17.78	**38.64**	20.87	7.21	21.52	42.20	29.16	8.53	25.12
Random Paired	5.09	4.04	81.06	17.78	9.09	22.20	3.39	16.59	24.77	18.16	44.03	22.38
w/ Group Attention	8.30	27.11	84.09	**82.22**	36.36	26.28	**26.85**	52.02	**44.95**	24.04	19.11	39.21
w/o *FlexMLP* ($ps = 16$)	5.86	13.75	**90.15**	59.26	29.54	29.02	8.79	40.81	30.28	30.18	45.05	34.79
Freeze Text Encoder	**12.02**	17.28	70.46	**82.22**	34.85	**39.48**	13.15	23.32	36.70	**42.20**	20.82	35.68
Llama3.2-1B Backbone	8.81	24.21	63.18	**82.22**	36.36	32.51	13.01	30.94	42.57	31.30	36.72	36.53
$\text{SLIP}_{\text{Small}}$ (40M)	11.27	9.84	43.18	32.59	36.36	33.01	14.67	22.42	20.18	20.46	**49.83**	26.71

Table 16: **Ablation Studies SFT performance.** Top-1 accuracy on four sensor QA benchmarks.

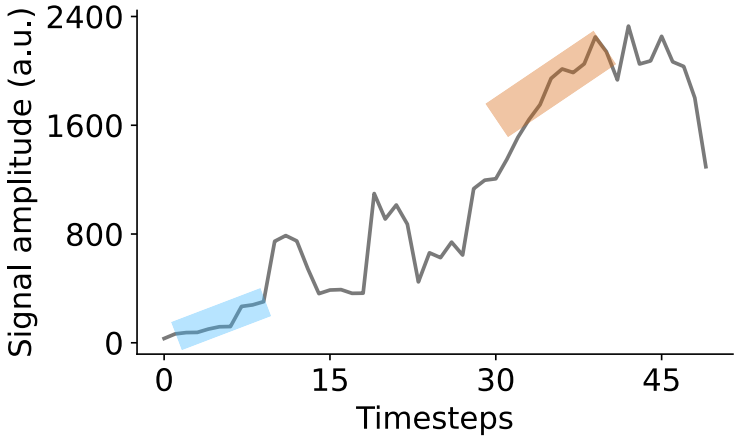
Model	TSQA	HAR-CoT	Sleep-CoT	ECG-QA-CoT	Avg
SLIP	**99.75**	**64.35**	74.19	37.18	68.87
Caption-Only	99.04	49.16	69.14	34.07	62.85
Contrastive-Only	95.88	48.57	29.57	38.21	53.06
Random Paired	74.40	27.93	21.51	33.16	39.25
w/ Group Attention	99.27	60.41	69.14	37.05	66.47
w/o *FlexMLP* ($ps = 16$)	98.65	61.66	69.03	38.17	66.88
Freeze Text Encoder	99.38	47.00	61.18	34.62	60.55
Llama3.2-1B Backbone	99.60	62.65	**74.25**	**43.47**	**69.99**
$\text{SLIP}_{\text{Small}}$ (40M)	99.35	50.58	64.09	25.70	59.93

Table 17: **Sensor captioning and question answering results with larger-LLM baselines.** Extends Table 4 with ITFormer-0.5B and ChatTS-8B, which use larger LLMs than the Gemma-3-270M backbone shared by OpenTSLM, SensorLM, and SLIP; they are shown in gray and excluded from highlighting. Metrics and protocol follow Table 4. Best results are in **bold** and second best are underlined.

Model	Sensor Captioning (M4)					TSQA	HAR-CoT	Sleep-CoT	ECG-QA-CoT
	BLEU@4	METEOR	ROUGE-L	SBERTSim	BERTScore	Accuracy	Accuracy	Accuracy	Accuracy
ITFormer	0.1641	0.3708	0.3182	0.8748	0.9001	89.35	41.29	43.99	38.01
ChatTS	0.1252	0.3518	0.2950	0.8500	0.8906	96.25	41.16	55.22	34.81
OpenTSLM-SP	0.0026	0.0456	0.0255	0.1551	0.7250	26.79	0.55	5.91	1.11
OpenTSLM-Flamingo	0.1141	0.3210	0.2894	0.7990	0.8858	78.12	63.43	68.49	35.50
SensorLM	0.2008	**0.4151**	0.3595	**0.9014**	0.9024	95.73	60.80	73.12	30.29
SLIP	**0.2046**	0.4027	**0.3658**	0.8977	**0.9077**	**99.75**	**64.35**	**74.19**	**37.18**

A.8 Additional Examples of Question Answering

Prompt:
You are an expert in time series analysis. Please generate a detailed caption for this time-series, describing it as accurately as possible



Generated Caption:
The time-series graph illustrates data points over a period of 50 units, showcasing a general upward trend with fluctuations. Initially, the series starts at a low point near 0 and gradually increases, reaching around 1000 by the 10th unit. A noticeable peak occurs around the 15th unit, followed by a decline, creating a trough near the 20th unit. From there, the series experiences a significant upward surge, peaking at approximately 2500 around the 35th unit. This is followed by minor fluctuations, maintaining a high level until the 45th unit...

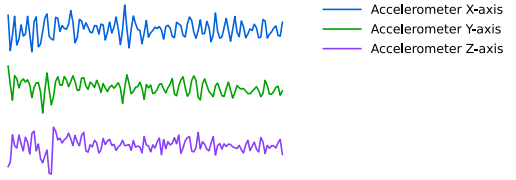
Figure 5: **A qualitative example of sensor captioning from the M4 dataset.** Highlighted segments denote salient local trends referenced in the generated caption.

Prompt:

```
Instructions:
 - Begin by analyzing the time series without assuming a
specific label.
 - Think step-by-step about what the observed patterns
suggest regarding movement intensity and behavior.
 - Write your rationale as a single, natural paragraph — do
not use bullet points, numbered steps, or section headings.
 - Do **not** mention any class label until the final
sentence.

 Possible activity labels are:
 biking, lying, running, sitting, standing, walking,
walking_down, walking_up.

 - Make sure that your last word is the answer. You MUST end
your response with "Answer: "
```



Prediction:

Rationale: The accelerometer data over the 2.56 second window shows significant variability across all three axes, indicating substantial movement. The **X-axis exhibits frequent peaks and troughs, suggesting lateral or forward-backward motion** that is consistent with dynamic activity. The **Y-axis also displays fluctuations**, though they are less pronounced, which might indicate less vertical movement or tilt. **The Z-axis shows moderate variability**, which could be associated with subtle changes in height or orientation. The overall pattern of the data **suggests a continuous and rhythmic motion, likely involving repetitive cycles** of movement. The presence of such variability and intensity across all axes is more characteristic of an activity involving consistent motion and balance adjustments rather than a static or minimal movement scenario. Therefore, the observed patterns are more aligned with the activity of biking. **Answer: biking.**

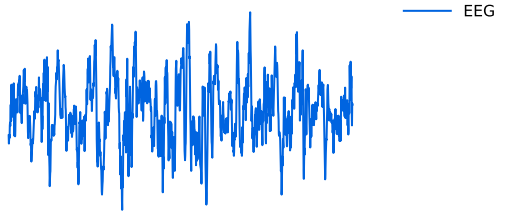
Ground Truth:

The accelerometer data over the 2.56 second window shows significant variability across all three axes, indicating substantial movement. The X-axis and Y-axis **exhibit frequent fluctuations with both positive and negative accelerations**, suggesting dynamic motion rather than a static position. The **Z-axis also shows variability, although with slightly less amplitude**, which could indicate vertical movement or changes in elevation. The overall pattern suggests a **repetitive, rhythmic motion consistent with a cycling activity,** where the body experiences continuous motion due to pedaling and shifts in balance. These characteristics are less likely to be observed in a stationary activity, which would typically show minimal changes in acceleration. Therefore, the observed patterns are more indicative of an active movement rather than stillness. **Answer: biking.**

Figure 6: A qualitative example of sensor question answering from the HAR-CoT dataset.

Prompt:

Instructions:
- Analyze the data objectively without presuming a particular label.
- Reason carefully and methodically about what the signal patterns suggest regarding sleep stage.
- Write your reasoning as a single, coherent paragraph. Do not use bullet points, lists, or section headers.
- Only reveal the correct class at the very end.
- Never state that you are uncertain or unable to classify the data. You must always provide a rationale and a final answer.Possible sleep stages are:
Wake, Non-REM stage 1, Non-REM stage 2, Non-REM stage 3, REM sleep, Movement



Prediction:

Rationale: The time series data presented shows a pattern characteristic of certain sleep stages, distinguished by specific EEG features. The signal displays a relatively consistent amplitude and frequency, with **occasional bursts of higher amplitude activity** that can be observed throughout the interval. These bursts are indicative of the presence of **sleep spindles**, which are a hallmark of a particular sleep stage. The overall pattern lacks the high-frequency, low-amplitude activity typically associated with wakefulness, and instead demonstrates the more rhythmic and stable activity seen during sleep. The presence of these features strongly suggests that the EEG data corresponds to a sleep stage characterized by reduced consciousness and decreased sensory perception. **Answer: Non-REM stage 2**

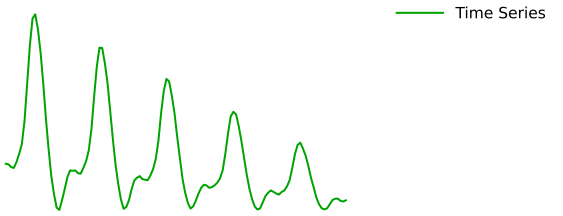
Ground Truth:

The time series data presented shows a pattern characteristic of certain sleep stages, distinguished by specific EEG features. The signal displays a rhythmic pattern with **periodic bursts of activity**, which may suggest the presence of **sleep spindles** typically associated with certain sleep stages. The amplitude and frequency of these bursts are consistent with known EEG patterns observed during sleep. Additionally, the overall variability and complexity of the signal, with its moderate amplitude fluctuations and rhythmic bursts, align with typical features observed during a particular stage of sleep. These characteristics help differentiate between the stages, and the presence of these specific patterns in the EEG data strongly indicates the sleep stage being observed. **Answer: Non-REM stage 2**

Figure 7: A qualitative example of sensor question answering from the Sleep-CoT dataset.

Prompt:

```
Instructions:
Given the following definitions: Constant volatility: The
time series shows relatively consistent fluctuation
magnitude throughout the period. Increased volatility: The
time series shows a rise in the magnitude of fluctuation
over time. Decreased volatility: The time series shows a
reduction in the magnitude of fluctuations over time. Select
one of the following answers that best describes the
provided time series:
(a) This time series has a constant volatility.
(b) This time series has an increased volatility.
(c) This time series has a decreased volatility.
Only answer (a), (b), or (c). Predict the Volatility Answer:
```



Prediction: C

Ground Truth: C

Figure 8: A qualitative example of sensor question answering from the TSQA dataset.

Prompt:

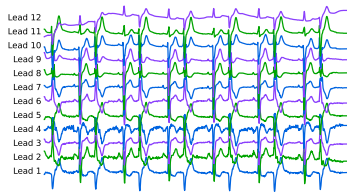
Clinical Context: 75-year-old male patient. 12-lead ECG. clinical recording. recorded with AT-6 C 5.5. Signal quality: baseline drift noted, static noise present, burst noise present, electrode artifacts present. extra beats detected during recording. pacemaker present.

Your task is to examine the ECG signal and answer the following medical question:

Question: Which diagnostic symptom does this ECG show, myocardial infarction in anterior leads or myocardial infarction in anteroseptal leads, excluding uncertain symptoms?

nstructions:
- Begin by analyzing the time series without assuming a specific answer.
- Think step-by-step about what the observed patterns suggest regarding the cardiac condition.
- Write your rationale as a single, natural paragraph — do not use bullet points, numbered steps, or section headings.
- Do **not** mention any final answer until the very end.
- Consider the ECG morphology, intervals, and any abnormalities that relate to the question.

Based on your analysis of the ECG data, select your answer from the following options:
complete left bundle branch block,
complete right bundle branch block,
digitalis effect, first degree av block,
...
...
...
- Make sure that your last word is the answer. You MUST end your response with "Answer: "



Prediction:

Side analysis for the 12-lead ECG of this 75-year-old male patient. The presence of a pacemaker and various artifacts such as baseline drift, static noise, burst noise, and electrode artifacts complicate the interpretation. However, focusing on the cardiac patterns, the leads V1 through V4, which represent the anteroseptal region, should be closely examined for signs of **myocardial infarction**. In these leads, significant ST-segment elevation or pathological Q waves would indicate myocardial infarction. The presence of extra beats may suggest arrhythmias, but the primary concern is the identification of ischemic changes. The absence of significant changes in the inferior leads (II, III, aVF) and **the lack of reciprocal changes in the anterior leads (V1-V3) further support the localization of the infarction to the anteroseptal region**. Given these observations, the ECG findings are consistent with myocardial infarction in the anteroseptal leads. Answer: myocardial infarction in anteroseptal leads. The ECG shows significant changes in the anteroseptal leads, specifically V1 to V4, which are indicative of myocardial infarction in this region. **Answer: myocardial infarction in anteroseptal leads.**

Ground Truth:

The ECG shows evidence of myocardial infarction, particularly focusing on the anterior leads. In this recording, the presence of ST-segment elevation in leads V1 through V4 is indicative of an anterior **myocardial infarction.** These leads correspond to the anterior wall of the heart, which is typically affected in such cases. **The baseline drift and noise present in the recording do not obscure the clear elevation of the ST segments in these leads.** The presence of a pacemaker and extra beats does not alter the interpretation of the ST-segment changes, which are crucial in diagnosing myocardial infarction. The absence of significant changes in the lateral leads (I, aVL, V5, V6) further supports the localization of the infarction to the anterior region rather than a lateral or subendocardial injury. Therefore, based on the observed patterns, the ECG is consistent with myocardial infarction in the anterior leads. **Answer: myocardial infarction in anterior leads.**

Figure 9: A qualitative failure example of sensor question answering from the ECG-QA-CoT dataset.