GestureLens: GENERALIZED LEARNING FRAMEWORK FOR IMU-BASED HUMAN GESTURE RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

IMU-based gesture interfaces are being increasingly adopted as efficient, accessible, and intuitive alternatives to traditional input methods, such as touchscreens and voice. However, current gesture recognition algorithms are tailored to work for specific devices (e.g., smartwatches vs. earbuds) or user populations (e.g., blind vs. sighted users), limiting their generalizability. In this paper, we design *GestureLens*, a generalized IMU-based gesture recognition framework that works across devices and populations with minimal training samples. To overcome the challenges and high cost of collecting large-scale labeled training data, *GestureLens* leverages readily available unlabeled human activity data. The *GestureLens* pipeline comprises two stages: (1) pre-training a motion representation model using abundant unlabeled human activity data, and (2) training a gesture classifier on top of the frozen representation using a small amount of labeled gesture data. For pre-training, we introduce a token-based strategy and embeddings that learn to identify and focus attention on the key motion signatures in the temporal data For classification, we design a text-guided classifier that can reliably differentiate between temporally or semantically similar gestures. We evaluate *GestureLens* across both hand gestures (captured through a smartwatch) and earbud gestures (captured through earbuds), using data collected from blind and sighted users. Across these diverse devices and user populations, *GestureLens* achieves an accuracy of 85%, across an average of 13 gesture classes using only 10% of labeled data for training. *GestureLens* significantly outperforms state-of-the-art self-supervised learning approaches and specialized gesture recognition models.

1 INTRODUCTION

Gesture interfaces are rapidly gaining popularity for their accessibility, efficiency, and intuitiveness in device interactions (Xu et al., 2022; Khanna et al., 2023; Wen et al., 2016a). Unlike traditional inputs like touchscreens, keyboards, or voice, gestures use natural body movements (Wobbrock et al., 2009; Lu et al., 2014). To capture and recognize these movements, researchers have explored sensing modalities including IMUs, cameras, acoustics, WiFi, and EMG. IMUs are already built into everyday devices such as phones, watches, and earbuds and can enable real-time gesture recognition on commodity wearables without requiring additional hardware (Honeywell International Inc., 2022). They are also robust to lighting conditions and occlusion (Pan et al., 2025) and incur low power and computational overhead (Xuan et al., 2024), making them well-suited for always-on, on-device gesture recognition. As a result, in this work, we focus on IMU-based gesture recognition.

There has been extensive research on IMU-based gesture recognition, but current systems require us to rebuild models from scratch for every new device and user demographic (Huang et al., 2021; Khanna et al., 2024; Wen et al., 2016a; Xu et al., 2020; Alkiek et al., 2023). These models require collecting extensive labeled data, which is time-consuming and expensive. Further, gesture recognition models trained for smartwatches do not work for earbud gestures, as they are tailored to capture hand movements and have not been trained to capture head movements. Similarly, gesture recognition models trained for sighted users have been shown not to work for blind users (Khanna et al., 2024; 2023). Therefore, the central question we ask in this work is:

Can we build IMU-based gesture recognition models that generalize across devices and populations without requiring extensive labeled data, while enabling real-time performance on commodity devices?

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.

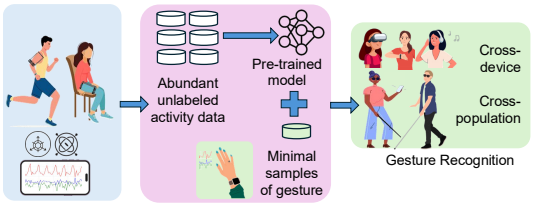

Figure 1: *GestureLens*: A framework enabling real-time robust gesture recognition across diverse devices and user populations.

Recent works in Human Activity Recognition (HAR) use self-supervised learning to achieve generalization across datasets, device placements, and domain shifts (Xu et al., 2021; Hong et al., 2024; Dai et al., 2024; Xu et al., 2023a). HAR models learn motion patterns from abundant unlabeled motion sensor data by masking segments of the motion signal and reconstructing them from surrounding context. Once learned from unlabeled data, these models require little labeled data for downstream tasks. The success of these techniques is enabled by the large amounts of activity data available (Stisen et al., 2015; Reyes-Ortiz et al., 2016; Malekzadeh et al., 2019). We want to apply the same self-supervised learning idea to gesture recognition. However, gesture data, even unlabeled, is not available at scale. Collecting gesture data requires users to repeatedly perform specific swipes or taps, which are not part of natural daily movements like walking. This makes it impractical to gather large datasets. However, we do have an abundance of unlabeled human activity data. Since both activities and gestures rely on the same basic movements and are limited by how the body can physically move, such as lifting or rotating a limb, we ask:

As a first step, we used existing self-supervised HAR techniques (Xu et al., 2021; 2023a; Dai et al., 2024) for pre-training and trained a gesture classifier on the resulting representations. However, our experiments yielded poor gesture recognition accuracy (Section §6.1). This shows that directly applying existing self-supervised techniques is not sufficient for gesture recognition. We identify three challenges that explain this mismatch:

(1) Activities are longer; Gestures are shorter. Activities are prolonged and continuous, so the motion that defines them is spread across many moments (e.g., multiple steps while walking). Gestures, in contrast, are brief and compact, with the key motion occurring in a very short time span (e.g., a single tap). If the short, defining part of a gesture is missed, there is little remaining information for the model to learn from.

(2) Activities are repetitive; Gestures are isolated. Activities are inherently repetitive. In walking, a step is always surrounded by other steps, creating a predictable cyclic pattern that helps the model learn. Gestures lack this context; a swipe is a single, isolated command. Models that rely on the redundancy found in continuous activities fail when applied to these singular, isolated events.

(3) Similar motions, different meanings. Many gestures have similar motion signals but distinct semantics. A "swipe up" and a "swipe down" share the same speed and energy profile, differing only in trajectory. Similarly, a "tap" and "double tap" share the same motion signal. Standard classifiers struggle to distinguish these fine-grained differences, especially when labeled data is limited.

To address these challenges, we present *GestureLens*, a generalized gesture recognition algorithm that works in two stages.

Stage 1: Token-based pre-training Recent work has explored tokenizing IMU signals into meaningful movement segments for self-supervised HAR (Tarale et al., 2026). However, these tokens are designed to capture the structure of continuous human activities. Our goal is different: gestures are short and isolated, so we identify the most critical portion of each motion, which we call the nucleus. The nucleus is the concentrated high-energy segment where the characteristic motion occurs (e.g., foot in-air phase while walking). We then mask 80% of the data within the nucleus and train the model to reconstruct these critical tokens. This forces the model to learn even short motion patterns rather than relying on surrounding context. In contrast, random masking used in existing works (Xu et al., 2021; Hong et al., 2024; Dai et al., 2024; Xu et al., 2023a) often misses the core part of the motion signal and thus fails to work for short gestures. By focusing on the nucleus, *GestureLens* addresses Challenges 1 and 2 by forcing the model to learn from the dense, isolated information that defines the gesture.

Stage 2: Text-Guided Classifier In the second stage, we pass target data through the frozen pre-trained model and obtain its learned motion embeddings, which are then fed into the classifier to predict gesture labels. The classifier is trained using a small amount of labeled samples. While HAR-based approaches typically use simple downstream classifiers, gesture recognition requires

Paradigm	System	Short Motion	Generalization	Real-time
Supervised	TapNet (Huang et al., 2021), EarBender* (Alkiek et al., 2023), Serendipity* (Wen et al., 2016a)	✓	✗	✗
	Khanna et al.* (Khanna et al., 2024)	✓	✗	✓
	DeepSense* (Yao et al., 2017)	✗	✓	✓
	IMUGPT 2.0* (Leng et al., 2024), GOAT (Miao & Chen, 2024), RAG-HAR (Sivaroopan et al., 2025)	✗	✓	✗
Masked SSL	LIMU-BERT* (Xu et al., 2021), UniHAR* (Xu et al., 2023a), Bio-PM* (Tarale et al., 2026)	✗	✓	✓
	CrossHAR (Hong et al., 2024), SelfPAB (Logacjov et al., 2024), Yuan et al. (Yuan et al., 2024)*, Qiu et al. (Qiu et al., 2025)*	✗	✓	✗
Contrastive SSL	ContrastSense* (Dai et al., 2024), Cosmo* (Ouyang et al., 2022)	✗	✓	✓
	ColloSSL* (Jain et al., 2022), CPCHAR (Haresamudram et al., 2021)	✗	✓	✗
GestureLens		✓	✓	✓

Table 1: Comparison of prior IMU-based gesture and activity recognition systems along: suitability for short-duration gesture, generalization across devices, and real-time feasibility, highlighting how *GestureLens* satisfies all three. SSL denotes Self-Supervised Learning. We compare *GestureLens* with state-of-the-art in each category, denoted by *

better differentiation due to motion similarity between gestures (Challenge 3). Our classifier uses contrastive learning plus a textual description of the gesture to differentiate between similar gestures. Traditional contrastive learning treats all negative pairs equally. To overcome this, we redesign the contrastive objective to enforce semantically-aware, non-uniform separation in the embedding space. We use textual descriptions to learn how much to move "tap" and "double-tap" from the "circle" gesture, and consequently learn to differentiate between similar-looking gestures. Importantly, the gesture description is only needed for training and does not have to be provided as input at runtime.

We evaluate *GestureLens* for gesture recognition accuracy across a diverse range of *4 public and 2 real-world datasets* designed to test its *cross-population* and *cross-form-factor* generalization capabilities. Specifically, we consider two distinct target populations (sighted and blind users) and different device form factors including earbuds and smartwatches. *GestureLens* utilizes a single pre-trained model across these diverse scenarios, requiring only 10% labeled data for classification, yet achieving an average accuracy of 85%. We conduct extensive comparisons with state-of-the-art works, including DeepSense (Yao et al., 2017), LIMU-BERT (Xu et al., 2021), UniHAR (Xu et al., 2023a), ContrastSense (Dai et al., 2024), Qiu et al. (Qiu et al., 2025), Yuan et al. (Yuan et al., 2024), ColloSSL (Jain et al., 2022), Cosmo (Ouyang et al., 2022) Bio-PM (Tarale et al., 2026), and specialized algorithms for blind users (Khanna et al., 2024) and earable gestures (Alkiek et al., 2023). *GestureLens* significantly outperforms these models across the evaluated datasets, with improvements of up to 50% on at least one dataset. Also, *GestureLens* is real-time and can run on smartphones, incurring an end-to-end latency of less than 67 msec.

2 Challenges in Generalized Gesture Recognition

The goal of *GestureLens* is to build an accurate IMU-based gesture recognition system that generalizes across devices and populations, while enabling real-time performance on commodity devices. Despite considerable work in human motion recognition (activities and gestures), generalized gesture recognition on resource-constrained devices remains an open problem. Table 1 shows the related work on human motion recognition that uses IMU sensors (the related work is expanded in Appendix G). These works have one or more of the following limitations—they are not designed to work for short motion durations (our evaluations show that when applied for short motion duration they do not work well); they do not generalize for different tasks; or they do not work in real-time on commodity devices.

Supervised learning methods build specialized models for individual devices or user populations, requiring substantial gesture data to generalize across settings. Masking-based self-supervised methods learn generalized motion representations from unlabeled HAR data, but their masking strategies are designed for continuous and repetitive activities rather than short, isolated gestures.

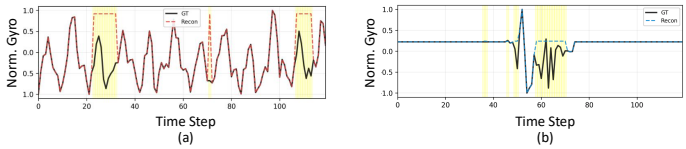

Figure 2: Random masking on continuous HAR and short-duration gestures.

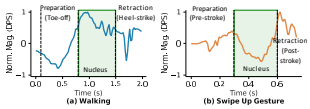

Figure 3: Three-phase structure of walking and swipe gestures.

Current works (Xu et al., 2021; 2023a; Hong et al., 2024) mask random segments anywhere in the signal and reconstruct them using surrounding temporal context. This works well for activities because information is distributed relatively evenly throughout repetitive motion patterns, but fails for gestures, where information is concentrated in a brief duration and the remaining signal is often generic preparation or retraction.

Figure 2 illustrates this difference. When masking 2 seconds in the middle of a walking sequence (a), the model observes before and after the masked region and can identify the activity context to predict the masked signal. For gestures (b), the same strategy provides little useful context because the gesture is short and isolated. If the random mask covers preparation or retraction, it removes little meaningful information; if it covers the gesture-defining region, the surrounding signal is insufficient for reconstruction. Contrastive-learning based methods improve cross-device and cross-dataset generalization, but existing approaches apply contrastive objectives during pre-training without gesture labels and therefore rely on motion similarity rather than semantic distinctions. We find that applying contrastive objectives during the fine-tuning stage is more effective, as class labels are available at this stage (see §4). We detail these design choices and related work in Appendix A.

3 Stage 1: Token-based Pre-training

In this section, we describe how *GestureLens* pre-trains a motion model on unlabeled HAR data that transfers to gesture recognition. The resulting encoder is then kept frozen; §4 describes how we train a classifier on the embeddings it produces.

Insight: Gestures are brief motion signals, and the informative part of the gesture that defines it lies in a very short duration of time. To learn short gestures effectively, models must focus on this most informative region. To learn from this informative region, we leverage the compositional nature of human motion (Yin & Davis, 2014; Benitez-Garcia et al., 2020; Khanna et al., 2023; Srivastava et al., 2022). Both gestures and activities are known to be compositional, where the motion can be divided into three phases: the pre-stroke phase (preparation), nucleus (the defining motion), and the post-stroke phase (retraction). Figure 3 illustrates this in two signals: (a) a walking activity and (b) a swipe-up gesture. In the case of walking, the actual activity is captured in part when the leg is in the air (swing: from leaving the ground to heel-strike). The portions before and after this can be thought of as getting into position and getting out of position. Similarly, for the "swipe up" gesture, the short 300 msec window shows when the gesture is performed. Pre-stroke is when the user positions their hand to perform the gesture, and post-stroke is when they get out of position. If we can learn the motion signature of this nucleus, then we can perform effective gesture recognition.

We therefore tokenize each motion window into these three phases and build pre-training around the most important token, the nucleus. Figure 4 shows the pipeline. We detect the nucleus in unlabeled activity data (§3.1), concentrate masking inside it (§3.2), and give the model two additional input encodings that mark the nucleus and the most discriminative sensor axis (§3.3). Appendix B.1 gives an overview of each component.

3.1 Nucleus Identification

The nucleus is the defining phase of a motion event. It is a brief burst of motion that distinguishes a motion event from another. In contrast to Natural Language Processing (NLP), where tokens (words) are discrete and clearly separated by spaces, IMU data is continuous. There are no explicit boundaries defining a nucleus in a continuous signal. Tokenizing continuous motion into meaningful units has recently been explored for self-supervised HAR. Bio-PM (Tarale et al., 2026) segments wrist accelerometer signals into movement units at acceleration zero-crossings, motivated by sub-movement theory from motor control, and uses each unit as a token for masked pre-training. This design segments the entire stream: each axis is tokenized independently, every segment becomes a token, and the method is built for long windows of continuous, free-living activity. Short gestures pose a different problem. A sub-second gesture produces only a handful of such segments; most

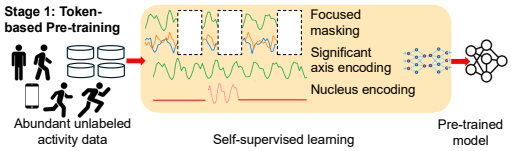

Figure 4: **Stage 1: Token-based pre-training**. The model learns from unlabeled activity data by applying focused masking to the nucleus (high-energy region), while nucleus and significant axis encodings guide attention to discriminative motion patterns.

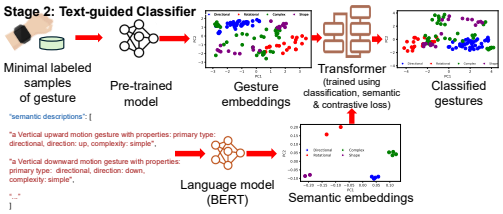

Figure 5: **Stage 2: Text-guided Contrastive Classifier**. Combines pre-trained motion embeddings with BERT semantic gesture descriptions. Uses semantic and contrastive losses to organize the space for accurate classification with minimal labels.

of the window is preparation, retraction, or stillness, and what matters is *which* segment carries the gesture's meaning. Rather than segmenting the whole signal into units of equal standing, we locate the single meaning-carrying token, the nucleus, and treat the phases around it as context.

We use the observation from Figure 3 that the nucleus has high energy compared to other parts of the motion (Khanna et al., 2023; 2024). We compute the energy of the signal as the intensity of motion across all sensor dimensions: $energy(t) = \sqrt{\sum_{i\in\{x,y,z\}}(acc_i(t)^2 + gyro_i(t)^2)}$. We then detect change points where the energy changes sharply over a window Δt, and define the nucleus as the contiguous segment between the first and last change point:

$$\mathcal{C} = \{t : |energy(t+\Delta t) - energy(t)| > motion_thresh\}, \qquad Nucleus = [\min \mathcal{C}, \max \mathcal{C}] \quad (1)$$

where $motion_thresh$ determines the detection sensitivity. Because nucleus duration (in samples) and magnitude vary with sampling rate, device, and user, we resample all signals to a common frequency and normalize the energy to $[0, 1]$, so that a single Δt and $motion_thresh$ apply across datasets. Detection therefore depends on when the signal transitions, not how large it is. Appendix B.2 details parameter selection and shows that the same parameters recover the nucleus across devices, populations, and sampling rates.

3.2 Focused Masking

Prior self-supervised HAR methods mask uniformly across the signal (Xu et al., 2021; 2023a), which is ineffective for short gestures where the informative content is concentrated in a brief segment. Instead, we place most masks inside the nucleus. Given a window of length T and an overall masking ratio ρ, we mask $m = \lceil \rho T \rceil$ positions, of which a fraction p_1 is drawn from the nucleus and the remainder from outside it:

$$|M \cap Nucleus| = \lceil p_1 m \rceil, \qquad |M \setminus Nucleus| = m - \lceil p_1 m \rceil \quad (2)$$

where M is the set of masked positions. This introduces a tradeoff. If p_1 is too low, most of the nucleus remains visible, and many masked samples fall in low-energy preparation or retraction; reconstruction becomes easy, and the model does not learn the core motion. If p_1 is too high, too little of the nucleus remains to anchor the reconstruction. We select p_1 on validation data. Table 5 shows performance peaks at $p_1 = 0.8$ across devices and populations. Appendix B.3 discusses this tradeoff and the construction of the nucleus mask in more detail.

Within the nucleus, we mask contiguous spans of 4–5 samples rather than single points. If the model observes an accelerating slope leading into a masked span, it cannot simply interpolate to the end; it must predict whether the motion continues accelerating or begins decelerating. It learns to reconstruct the *shape* of the motion rather than the continuity of the line (Figure 2). The pre-training objective is to reconstruct the motion data in the masked positions: $\mathcal{L}_{pretrain} = \sum_{t\in M}(x_t - \hat{x}_t)^2$ where $\hat{x}_t$ is the reconstructed output.

3.3 Input Encodings

To supplement focused masking, we provide two additional inputs that mark the nucleus boundaries and identify the IMU axis most relevant to the motion.

Nucleus Encoding: We define a binary mask $M_{nucleus}(t) = 1$ if $t \in Nucleus$ (Eq. 1) and 0 otherwise. It is mapped to a learnable vector. This lets self-attention amplify focus on the nucleus and attend less to preparation and retraction phases of the motion. Without this guidance, attention

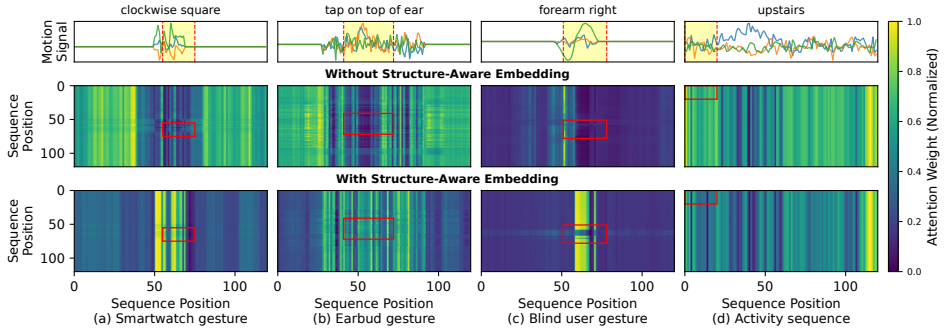

Figure 6: Attention maps with and without token-based pre-training across (a) smartwatch gestures, (b) earbud gestures, (c) blind-user gestures, and (d) an activity sequence. Red rectangles mark the detected nucleus. With token-based pre-training, attention concentrates in vertical bands aligned with the nucleus; without it, attention is dispersed.

disperses uniformly across the window. Figure 6 compares attention maps from a model pre-trained with and without our nucleus, on smartwatch gestures, earbud gestures, blind-user gestures, and stair climbing. Without it, attention is spread almost uniformly across the window. With it, attention concentrates in bands aligned with the detected nucleus, across all four device and population settings, and reconstruction MSE drops by 8.7%. Appendix B.5 gives the extended analysis.

Significant Axis Encoding: Even with guided attention, directional ambiguity remains. Walking upstairs and downstairs produce similar step patterns except along the gravitational axis. Swipe left and swipe right differ only in direction. To find which axis matters, we look at the gyroscope inside the nucleus and pick the axis with the most rotation. We mark this axis with a 1 and the others with a 0, and map this to a learnable vector. This lets us encode direction alongside temporal dynamics. Without it, the model tends to learn direction-invariant features that cannot separate opposing movements. The input to the model combines these with a standard positional encoding (Xu et al., 2021): $E = E_{input} + E_{position} + E_{nucleus} + E_{sig_axis}$, where E_{input} is a linear projection of the raw IMU values. Further details of each encoding are in Appendix B.4.

4 Stage 2: Text-Guided Classifier

In Stage 2, we use a small amount of labeled data to train a gesture classifier on top of the pre-trained model. The Stage-1 encoder is kept frozen: every labeled gesture sample is passed through it to obtain motion embeddings, and only the classifier is trained on these embeddings.

Insight: The pre-trained model captures motion patterns, but it lacks an understanding of gesture semantics. While activities in HAR are often distinct, many gestures share nearly identical signals that simple classifiers struggle to separate, especially when gesture data is limited. For example, a *tap* and *double-tap* produce similar signal peaks despite representing different actions. However, their semantic difference can guide the classifier to separate them in the embedding space based on not only the motion signal but also their semantics.

To this end, we use a text-guided contrastive classifier that combines motion patterns and gesture semantics. Unlike contrastive pre-training methods (§2), which have no labels and must define positives and negatives from motion similarity alone, we apply contrastive learning at the classification stage, where class labels and class semantics are available.

Figure 5 shows the gesture classification pipeline. We pass labeled target data through the frozen pre-trained encoder to obtain motion embeddings, assign each gesture class a short text description (§4), and train the classifier with an objective that uses these descriptions to shape the embedding space (§4). Text descriptions are used only during training and are not needed at inference. Appendix C.1 gives further details.

Text Descriptions Each class in the target dataset is assigned a text description using a structured template that specifies the class' category, orientation, and complexity level: "a <direction> <type> gesture with properties: primary type: <category>, direction <orientation>, complexity: <complexity-level>" For instance, the description for *swipe up* is: "an upward swipe gesture with properties: primary type: directional, direction: up, complexity: simple." We encode each description with a BERT language model, giving one fixed text embedding per class. Descriptions with similar meanings map to nearby points, so the embeddings capture relationships between gestures: *swipe up* and *swipe down* are related directional gestures, while *circle* and *figure-eight* share shape-

Dataset	Sensors	Classes	Users	Samples
Sony Watch Gesture (Porzi et al., 2013) (Publicly available)	A	20	8	3200
UMAHand (Casilari et al., 2024) (Publicly available)	A,G,M	4	25	102
UTD-MHAD (Chen et al., 2015) (Publicly available)	A,G,V	8	4	256
MMG Gesture (Abbas et al., 2025) (Publicly available)	A,G	11	43	23640
Sighted user Earbud Gesture (Custom)	A,G	7	9	315
Sighted user Watch Gesture (Custom)	A,G	12	10	1200
HHAR (Stisen et al., 2015) (Publicly available)	A,G	6	9	9166
UCI (Reyes-Ortiz et al., 2016) (Publicly available)	A,G	6	30	2088
Motion (Malekzadeh et al., 2019) (Publicly available)	A,G	6	24	4534
Shoaib (Shoaib et al., 2014) (Publicly available)	A,G,M	7	10	10500

Table 2: Summary of datasets used for evaluating *GestureLens*. A=accelerometer, G=gyroscope, M=magnetometer, V = video

tracing properties. Descriptions are written once per class and embedded once, so they add no cost during training or inference (Appendix C.2). Appendix 6.2 shows that performance is stable across different description styles, and drops when descriptions are shuffled or replaced with non-semantic anchors.

Contrastive Learning with Text Guidance To distinguish gestures with similar motion patterns, we employ a contrastive learning objective. The key contribution lies in setting contrastive distances from the semantic relationships between text descriptions, rather than treating all negative pairs uniformly. For a negative pair (samples from different gesture classes), the required separation depends on the semantic distance between the two classes' text embeddings. This ensures that physically similar but semantically distinct gestures like *swipe up* versus *swipe down* maintain appropriate separation in the embedding space. We combine three loss components: (1) **classification loss** $\mathcal{L}_{\text{class}}$ for basic boundaries, (2) **semantic loss** $\mathcal{L}_{\text{sem}} = \sum_{i,j} w_{ij} \max(0, m_{ij} - d(f_i, f_j))$ to enforce semantic separation, and (3) **contrastive loss** $\mathcal{L}_{\text{con}} = -\sum_i \log \frac{\exp(z_i^T z_{\text{same}}/\tau)}{\sum_{j \neq i} \exp(z_i^T z_j/\tau)}$ to cluster similar gestures. The final objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + w_s \mathcal{L}_{\text{sem}} + w_c \mathcal{L}_{\text{con}}$, where w_s and w_c increase gradually during training (see Appendix C.3).

5 Evaluation Setup

In this section, we detail the dataset and metrics we use to evaluate *GestureLens*. Per-dataset collection details and baseline details are in Appendix E.

We evaluate *GestureLens* on four public and two custom-collected gesture datasets, spanning two user populations (sighted and blind users) and two wearable form factors (smartwatch and earbud), and on four public HAR datasets. Table 2 summarizes all datasets. Our data collection was conducted following approval from our institute's Institutional Review Board (IRB).

Token-based pre-training uses only the unlabeled HHAR activity dataset. The pre-training corpus is disjoint from the gesture evaluation datasets and contains no gesture labels. A single pre-trained encoder is used for every result in this paper; it is frozen thereafter. For each target dataset, we train only the classifier, using a stratified 10% subset of the labeled data. The remaining labeled data of each target dataset is held out and used for evaluation. Training and evaluation sets are disjoint, and the pre-training corpus is disjoint from every evaluation dataset in users, devices, sampling rates, and classes. Baselines are evaluated under the same conditions: we take each baseline's pre-trained encoder, train its classifier on the identical 10% split with hyperparameter tuning, and evaluate on the identical held-out set.

Generalization axes. The evaluation is designed to test five generalizations:

- The blind user dataset represents a challenging test case, as blind users exhibit higher motion variability than sighted users due to a lack of visual feedback during training (Khanna et al., 2024). Successfully recognizing gestures from this population demonstrates the robustness of *GestureLens* to user variance beyond typical inter-user variability.
- We evaluate across three distinct form factors: earbuds, smartphones, and two types of smartwatches (Fossil Gen 5, Sony SmartWatch) with different IMU characteristics, sampling rates (10–200 Hz), and wearing positions, showing performance across heterogeneous hardware.
- The earbud dataset, with seven gestures and limited labeled samples, evaluates whether our text-guided classification enables learning from minimal data, where collecting extensive labeled datasets is infeasible.

	Sony Watch SU Hand Gesture (P)	**UMAHand** (P)	**UTD-MHAD** (P)	**MMG Gesture** (P)	**SU Earbud Gesture (C)**	**BU Hand Gesture (C)**
Generalizable gesture recognition methods						
DeepSense (Yao et al., 2017)	0.67 / 0.67	0.48 / 0.42	0.20 / 0.22	0.33 / 0.30	0.23 / 0.22	0.52 / 0.53
LIMU-BERT (Xu et al., 2021)	0.60 / 0.62	0.32 / 0.28	0.48 / 0.42	0.33 / 0.30	0.42 / 0.40	0.56 / 0.58
ColloSSL (Jain et al., 2022)	0.77 / 0.76	0.48 / 0.42	0.36 / 0.33	0.64 / 0.60	0.33 / 0.29	0.36 / 0.33
Cosmo (Ouyang et al., 2022)	**0.91** / 0.84	0.68 / 0.66	0.45 / 0.43	0.71 / 0.64	0.60 / 0.57	0.65 / 0.61
UniHAR (Xu et al., 2023a)	0.76 / 0.77	0.40 / 0.37	0.60 / 0.55	0.28 / 0.28	0.37 / 0.36	0.30 / 0.29
ContrastSense (Dai et al., 2024)	0.79 / 0.79	0.49 / 0.45	0.39 / 0.37	0.37 / 0.37	0.30 / 0.29	0.58 / 0.55
Yuan et al. (Yuan et al., 2024)	0.78 / 0.74	0.77 / 0.75	0.73 / 0.67	**0.79 / 0.75**	0.53 / 0.51	0.70 / 0.68
Qiu et al. (Qiu et al., 2025)	0.59 / 0.58	0.39 / 0.38	0.77 / 0.75	0.68 / 0.61	0.48 / 0.48	0.50 / 0.44
Bio-PM (Tarale et al., 2026)	**0.91 / 0.88**	0.61 / 0.62	0.55 / 0.50	0.43 / 0.42	0.66 / 0.88	0.45 / 0.52
Dataset-specific gesture recognition methods						
Serendipity (Wen et al., 2016b)	0.66 / 0.69	–	–	–	–	–
EarBender (Alkiek et al., 2023)	–	–	–	–	0.32 / 0.31	–
Khanna et al. (Khanna et al., 2024)	–	–	–	–	–	0.60 / 0.62
GestureLens	0.86 / 0.85	**0.88 / 0.85**	**0.79 / 0.78**	0.77 / **0.75**	**0.84 / 0.85**	**0.85 / 0.85**

Table 3: Performance comparison of *GestureLens* and baselines across datasets. Each entry reports Accuracy / F1-score. SU: Sighted User, BU: Blind User, P: Publicly available, C: Custom collected.

- Beyond gesture recognition, we evaluate *GestureLens* on HAR benchmarks to demonstrate its expandability to human motion beyond gestures. This approach achieves HAR performance on par with or better than recent systems, validating that token-based pre-training generalizes across different motion types.
- To assess real-time performance, we implement the gesture classifier on a commodity smartphone (Samsung Galaxy S20) and evaluate on-device inference performance, showing that *GestureLens* can operate under practical latency and resource constraints.

While we acknowledge that these datasets remain controlled compared to unconstrained real-world scenarios, they represent the standard evaluation setting in gesture recognition research. Our contribution focuses on moving from random masking to nucleus identification and focused masking, evaluated using benchmarks that enable direct comparison with prior work. In the future, we will explore more diverse real-world deployments. Next, we detail our datasets used for evaluation.

6 Results

In this section, we present a comprehensive evaluation of *GestureLens* for gesture recognition for cross-form-factor and cross-population setups, on publicly available and custom-collected datasets. We first summarize our key findings.

- *GestureLens* achieves >85% accuracy in gesture recognition across two wearable form factors (smartwatch, earbuds), two user populations (blind, sighted), and datasets with > 40 users. The second-best-performing baseline performs 50% worse on at least one dataset.
- *GestureLens* acts as a generalized model, outperforming recognition algorithms specially designed for different form factors or user populations. It achieves 85% earable gestures (surpassing EarBender (Alkiek et al., 2023)) and, for blind user gestures (surpassing Khanna et al. (Khanna et al., 2024)), demonstrates strong generalization with limited labeled data.
- *GestureLens* not only achieves high accuracy for the gesture recognition task, but also for the activity recognition task, even though *GestureLens* was not designed for activity recognition. *GestureLens* achieves a similar accuracy (93%) to state-of-the-art activity recognition baselines.
- *GestureLens* can efficiently run on smartphones. It only incurs a low end-to-end latency of 67 msec for gesture recognition.

6.1 Gesture Recognition Across Populations and Form Factors

Table 3 shows the accuracy and F1-score of *GestureLens* and the baselines. On the *Sighted user-hand gesture* dataset, *GestureLens* achieves 86% accuracy, outperforming DeepSense (67%) and LIMU-BERT (60%). UniHAR and ContrastSense achieve 76% and 79%, respectively, while Cosmo and Bio-PM achieve the highest accuracy 91%. However, the performance of these methods varies substantially across datasets, particularly across different form factors and user populations. We also evaluated IMUGPT 2.0 (Leng et al., 2024) for smartwatch gesture recognition; however, it consistently underperformed all generalization-focused baselines (41% accuracy). As a result, we

Token-based Pre-training		Text-guided Classifier		Performance	
Nucleus Enc.	Sig. Axis Enc.	Contrastive Loss	Semantic Loss	Accuracy	F1
✗	✗	✓	✓	0.71, 0.65, 0.73	0.71, 0.65, 0.73
✗	✓	✓	✓	0.77, 0.68, 0.77	0.76, 0.69, 0.75
✓	✗	✓	✓	0.78, 0.70, 0.80	0.76, 0.68, 0.80
✓	✓	✗	✗	0.79, 0.58, 0.73	0.78, 0.56, 0.73
✓	✓	✗	✓	0.80, 0.75, 0.79	0.79, 0.76, 0.79
✓	✓	✓	✗	0.80, 0.67, 0.79	0.79, 0.64, 0.80
GestureLens (Full Model)				**0.86, 0.85, 0.85**	**0.85, 0.84, 0.85**

Table 4: Ablation study isolating the contribution of token-based pre-training and classifier design. Accuracy and F1 are reported as: Hand (SU), Earbud (SU), Hand (BU).

omit IMUGPT 2.0 from subsequent cross-form-factor and cross-population evaluations. On *UMA-Hand*, *GestureLens* achieves 88% accuracy and 85% F1-score, outperforming all baselines. On *UTD-MHAD*, *GestureLens* achieves 79% accuracy and 78% F1-score, again outperforming the baselines. On *MMG Gesture*, *GestureLens* achieves 77% accuracy, close to Yuan et al. (Yuan et al., 2024) (79%).

For the *Sighted user-earbud gesture* dataset, *GestureLens* achieves 85% accuracy, while the baseline accuracies drop substantially (DeepSense: 22%, LIMU-BERT: 42%, UniHAR: 39%, ContrastSense: 30%). The earbud gestures are short and subtle, and the dataset contains limited labeled data, making them challenging for existing methods. In contrast, *GestureLens* maintains high accuracy with only 10% labeled data. On the *Blind user-hand gesture* dataset, *GestureLens* achieves 85% accuracy, outperforming all baselines; ContrastSense is the strongest baseline at 59%.

6.2 Ablation Study

We conduct an ablation study to analyze the contributions of token-based pre-training and classifier design, reporting accuracy and F1-score in Table 4. We keep the architecture fixed across all ablations and selectively remove the proposed encodings and classifier objectives.

Impact of pre-training encodings. Removing both nucleus and significant-axis encodings consistently reduces performance across datasets. Adding either encoding improves performance, with comparable gains from each. Nucleus encoding focuses the model on the temporally informative region, while significant-axis encoding captures directional information in the IMU signal, indicating that the two provide complementary information.

Impact of classifier objectives. Starting from classification loss alone, adding either semantic or contrastive loss improves performance, particularly for the subtle and similar earbud gestures. The full objective performs best, showing that both semantic guidance and contrastive learning contribute to separating motion-similar gestures. We further compare the full *GestureLens* model trained with cross-entropy (CE) alone against the complete text-guided objective. The text-guided objective improves performance from 0.79/0.78 to 0.86/0.85 on SU hand gestures, 0.71/0.70 to 0.84/0.83 on SU earbud gestures, and 0.73/0.73 to 0.85/0.85 on BU hand gestures (Accuracy/F1). These results show consistent benefits across form factors and populations. Additional ablations on nucleus masking ratios and textual descriptors are reported in Appendix F.5.

7 Conclusion

In this paper, we introduce *GestureLens*, a generalized learning framework for human gesture recognition across form factors and user populations. We use a two-stage approach: token-based pre-training on unlabeled activity data and a text-guided classifier trained with limited labeled gesture data. Token-based pre-training focuses on the most discriminative motion regions, while the text-guided classifier separates semantically similar gestures. Rather than directly applying existing self-supervised and contrastive learning methods, we adapt both stages to the short, isolated, and fine-grained nature of gestures. We introduce nucleus-based tokenization for pre-training and a semantically-aware, non-uniform contrastive objective for classification, enabling *GestureLens* to capture the motion patterns that define gestures. We evaluate *GestureLens* across smartwatch and earbud devices, sighted and blind users, and activity and gesture tasks. It achieves 85% accuracy using only 10% of labeled data and consistently outperforms existing gesture recognition and self-supervised methods. Our results demonstrate that a unified real-time model can generalize across diverse devices and user populations, reducing the need for device- or population-specific models.

AI USE STATEMENT

We used generative AI tools, including GitHub Copilot and LLMs, for coding assistance and writing assistance. GitHub Copilot was used to generate and accelerate the implementation of code based on instructions provided by the authors. LLMs were also used to assist with writing by improving grammar, polishing text, and shortening or restructuring existing author-written text. No research ideas, experimental designs, scientific claims, analyses, or results were generated by or derived from LLMs. All AI-assisted code and text were reviewed by the authors, and code was tested as part of the experimental workflow. The authors take responsibility for the final content of this work, including all text, claims, code, and experimental results.

ETHICS STATEMENT

This work involves human-subject data collected under an approved Institutional Review Board (IRB) protocol. All data collection and experimental procedures were conducted in accordance with the approved protocol. The study involves wearable sensor data for human gesture recognition, and we took appropriate measures to conduct the study and handle the collected data in accordance with the approved research procedures.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, pre-processing, model architecture, training procedures, evaluation protocols, and experimental settings in the main paper and appendix. Additional implementation details needed to reproduce our experiments are provided in the appendix. Upon acceptance, we will release the source code and trained model weights to facilitate reproduction of our results.

REFERENCES

URL https://pytorch.org/mobile/home/.

Khalid A Abbas, Mofeed Turky Rashid, and Luigi Fortuna. The mechanomyographic dataset of hand gestures harvested using an accelerometer and gyroscope. *Data in Brief*, 60:111558, 2025.

Muneer Al-Hammadi, Ghulam Muhammad, Wadood Abdul, Mansour Alsulaiman, and M Shamim Hossain. Hand gesture recognition using 3d-cnn model. *IEEE Consumer Electronics Magazine*, 9(1):95–101, 2019.

Khaled Alkiek, Khaled A Harras, and Moustafa Youssef. Eargest: Hand gesture recognition with earables. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 91–99. IEEE, 2022.

Khaled Alkiek, Moustafa Youssef, and Khaled A Harras. Earbender: Enabling rich imu-based natural hand-to-ear interaction in commodity earables. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*, pp. 333–338, 2023.

Shiri Azenkot and Nicole B Lee. Exploring the use of speech input by blind people on mobile devices. In *Proceedings of the 15th international ACM SIGACCESS conference on computers and accessibility*, pp. 1–8, 2013.

Gibran Benitez-Garcia, Muhammad Haris, Yoshiyuki Tsuda, and Norimichi Ukita. Continuous finger gesture spotting and recognition based on similarities between start and end frames. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):296–307, 2020.

Eduardo Casilari, Jennifer Barbosa-Galeano, and Francisco J González-Cañete. Umahand: A dataset of inertial signals of typical hand activities. *Data in Brief*, 55:110731, 2024.

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

Jiangcheng Chen, Sheng Bi, George Zhang, and Guangzhong Cao. High-density surface emg-based gesture recognition using a 3d convolutional neural network. *Sensors*, 20(4):1201, 2020.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.

Hung-Yuan Chung, Yao-Liang Chung, and Wei-Feng Tsai. An efficient hand gesture recognition system based on deep cnn. In *2019 IEEE International Conference on Industrial Technology (ICIT)*, pp. 853–858. IEEE, 2019.

Gaole Dai, Huatao Xu, Hyungjun Yoon, Mo Li, Rui Tan, and Sung-Ju Lee. Contrastsense: Domain-invariant contrastive learning for in-the-wild wearable sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–32, 2024.

Amit Das, Ivan Tashev, and Shoaib Mohammed. Ultrasound based gesture recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 406–410. IEEE, 2017.

Harish Haresamudram, Irfan Essa, and Thomas Plötz. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–26, 2021.

Honeywell International Inc. Tiny devices, big impact: How inertial measurement units power everyday life, December 2022. URL https://www.honeywell.com/us/en/news/2022/12/tiny-devices-big-impact-how-inertial-measurement-units-power-everyday-life. Accessed: 2026-01-29.

Zhiqing Hong, Zelong Li, Shuxin Zhong, Wenjun Lyu, Haotian Wang, Yi Ding, Tian He, and Desheng Zhang. Crosshar: Generalizing cross-dataset human activity recognition via hierarchical self-supervised pretraining. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(2):1–26, 2024.

Rong Hu, Ling Chen, Shenghuan Miao, and Xing Tang. Swl-adapt: An unsupervised domain adaptation model with sample weight learning for cross-user wearable human activity recognition. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 37, pp. 6012–6020, 2023.

Michael Xuelin Huang, Yang Li, Nazneen Nazneen, Alexander Chao, and Shumin Zhai. Tapnet: The design, training, implementation, and applications of a multi-task learning cnn for off-screen mobile input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2021.

Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. Collossl: Collaborative self-supervised learning for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–28, 2022.

Shaun K Kane, Jacob O Wobbrock, and Richard E Ladner. Usable gestures for blind people: understanding preference and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 413–422, 2011.

Peiqi Kang, Jinxuan Li, Bingfei Fan, Shuo Jiang, and Peter B Shull. Wrist-worn hand gesture recognition while walking via transfer learning. *IEEE Journal of Biomedical and Health Informatics*, 26(3):952–961, 2021.

Fahim Kawsar, Chulhong Min, Akhil Mathur, and Alessandro Montanari. Earables for personal-scale behavior analytics. *IEEE Pervasive Computing*, 17(3):83–89, 2018. doi: 10.1109/MPRV.2018.03367740.

Prerna Khanna, Shirin Feiz, Jian Xu, IV Ramakrishnan, Shubham Jain, Xiaojun Bi, and Aruna Balasubramanian. Accesswear: Making smartphone applications accessible to blind users. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–16, 2023.

Prerna Khanna, IV Ramakrishnan, Shubham Jain, Xiaojun Bi, and Aruna Balasubramanian. Hand gesture recognition for blind users by tracking 3d gesture trajectory. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2024.

Donald Richard John Laming. Information theory of choice-reaction times. 1968.

Zikang Leng, Amitrajit Bhattacharjee, Hrudhai Rajasekhar, Lizhe Zhang, Elizabeth Bruda, Hyeokhyen Kwon, and Thomas Plötz. Imugpt 2.0: Language-based cross modality transfer for sensor-based human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(3):1–32, 2024.

Barbara Leporini, Maria Claudia Buzzi, and Marina Buzzi. Interacting with mobile devices via voiceover: usability and accessibility issues. In *Proceedings of the 24th Australian computer-human interaction conference*, pp. 339–348, 2012.

Aleksej Logacjov, Sverre Herland, Astrid Ustad, and Kerstin Bach. Selfpab: large-scale pre-training on accelerometer data for human activity recognition. *Applied Intelligence*, 54(6):4545–4563, 2024.

Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE transactions on human-machine systems*, 44(2):293–299, 2014.

Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, pp. 49–58, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362832. doi: 10.1145/3302505.3310068. URL https://doi.org/10.1145/3302505.3310068.

Shenghuan Miao and Ling Chen. Goat: A generalized cross-dataset activity recognition framework with natural language supervision. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–28, 2024.

Markus Modzelewski and Esteban Bayro Kaiser. Hand gesture recognition interface for visually impaired and blind people. In *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, pp. 201–206. IEEE, 2012.

Aparna Mohanty, Sai Saketh Rambhatla, and Rajiv Ranjan Sahay. Deep gesture: static hand gesture recognition using cnn. In *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 2*, pp. 449–461. Springer, 2017.

GRS Murthy and RS Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management*, 2(2):405–410, 2009.

Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, 6(8):73, 2020.

Xiaomin Ouyang, Xian Shuai, Jiayu Zhou, Ivy Wang Shi, Zhiyuan Xie, Guoliang Xing, and Jianwei Huang. Cosmo: contrastive fusion learning with small data for multimodal human activity recognition. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pp. 324–337, 2022.

Shaohua Pan, Xinyu Yi, Yan Zhou, Weihua Jian, Yuan Zhang, Pengfei Wan, and Feng Xu. Diffcap: Diffusion-based real-time human motion capture using sparse imus and a monocular camera. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

Lorenzo Porzi, Stefano Messelodi, Carla Mara Modena, and Elisa Ricci. A smart watch-based gesture recognition system for assisting people with visual impairments. In *Proceedings of the 3rd ACM International Workshop on Interactive Multimedia on Mobile & Portable Devices*, IMMPD '13, pp. 19–24, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323994. doi: 10.1145/2505483.2505487. URL https://doi.org/10.1145/2505483.2505487.

Kolla Bhanu Prakash, Rama Krishna Eluri, Nalluri Brahma Naidu, Sri Hari Nallamala, Pragyaban Mishra, and P Dharani. Accurate hand gesture recognition using cnn and rnn approaches. *International Journal*, 9(3), 2020.

Wanbin Qi, Ronghui Zhang, Jiaen Zhou, Hao Zhang, Yanxi Xie, and Xiaojun Jing. A resource-efficient cross-domain sensing method for device-free gesture recognition with federated transfer learning. *IEEE Transactions on Green Communications and Networking*, 7(1):393–400, 2023.

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.

Hanae Rateau, Edward Lank, and Zhe Liu. Leveraging smartwatch and earbuds gesture capture to support wearable interaction. *Proceedings of the ACM on Human-Computer Interaction*, 6(ISS): 31–50, 2022.

Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomput.*, 171(C):754–767, January 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2015.07.085. URL https://doi.org/10.1016/j.neucom.2015.07.085.

Andrea Rosales Sanabria, Franco Zambonelli, and Juan Ye. Unsupervised domain adaptation in activity recognition: A gan-based approach. *IEEE Access*, 9:19421–19438, 2021.

Soroush Shahi, Vimal Mollyn, Cori Tymoszek Park, Runchang Kang, Asaf Liberman, Oron Levy, Jun Gong, Abdelkareem Bedri, and Gierad Laput. Vision-based hand gesture customization from a single demonstration. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–14, 2024.

Sungho Shin and Wonyong Sung. Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2274–2277. IEEE, 2016.

Muhammad Shoaib, Silvia Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014. doi: 10.3390/s140610146.

Nirhoshan Sivaroopan, Hansi Karunarathna, Chamara Madarasingha, Anura Jayasumana, and Kanchana Thilakarathna. Rag-har: Retrieval augmented generation-based human activity recognition. *arXiv preprint arXiv:2512.08984*, 2025.

Elnaz Soleimani and Ehsan Nazerfard. Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing*, 426:26–34, 2021.

Tanmay Srivastava, Prerna Khanna, Shijia Pan, Phuc Nguyen, and Shubham Jain. Muteit: Jaw motion based unvoiced command recognition using earable. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–26, 2022.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pp. 127–140, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336314. doi: 10.1145/2809695.2809718. URL https://doi.org/10.1145/2809695.2809718.

Prithviraj Tarale, Kiet Chu, Abhishek Varghese, Kai-Chun Liu, Maxwell A Xu, Mohit Iyyer, and Sunghoon Lee. Bio-inspired self-supervised learning for wrist-worn accelerometer data. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=o4TAPjQnI6.

Zhengjie Wang, Yushan Hou, Kangkang Jiang, Wenwen Dou, Chengming Zhang, Zehua Huang, and Yinjing Guo. Hand gesture recognition based on active ultrasonic sensing of smartphone: a survey. *IEEE Access*, 7:111897–111922, 2019.

Hiroki Watanabe and Tsutomu Terada. Improving ultrasound-based gesture recognition using a partially shielded single microphone. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pp. 9–16, 2018.

Wentao Wei, Qingfeng Dai, Yongkang Wong, Yu Hu, Mohan Kankanhalli, and Weidong Geng. Surface-electromyography-based gesture recognition by multi-view deep learning. *IEEE Transactions on Biomedical Engineering*, 66(10):2964–2973, 2019.

Hongyi Wen, Julian Ramos Rojas, and Anind K Dey. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pp. 3847–3851, 2016a.

Hongyi Wen, Julian Ramos Rojas, and Anind K Dey. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pp. 3847–3851, 2016b.

Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1083–1092, 2009.

Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pp. 220–233, 2021.

Huatao Xu, Pengfei Zhou, Rui Tan, and Mo Li. Practically adopting human activity recognition. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–15, 2023a.

Huatao Xu, Pengfei Zhou, Rui Tan, and Mo Li. Practically adopting human activity recognition. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–15, 2023b.

Jian Xu, Syed Masum Billah, Roy Shilkrot, and Aruna Balasubramanian. Darkreader: bridging the gap between perception and reality of power consumption in smartphones for blind users. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 96–104, 2019.

Xuhai Xu, Haitian Shi, Xin Yi, WenJia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K Dey. Earbuddy: Enabling on-face interaction via wireless earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020.

Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, et al. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–19, 2022.

Jiawei Xuan, Ting Zhu, Gao Peng, Fayou Sun, and Dawei Dong. A review on the inertial measurement unit array of microelectromechanical systems. *Sensors*, 24(22):7140, 2024.

Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*, pp. 351–360, 2017.

Ying Yin and Randall Davis. Real-time continuous gesture recognition for natural human-computer interaction. In *2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 113–120. IEEE, 2014.

Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, Aidan Acquah, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *NPJ digital medicine*, 7(1):91, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Peide Zhu, Hao Zhou, Shumin Cao, Panlong Yang, and Shuangshuang Xue. Control with gestures: A hand gesture recognition system using off-the-shelf smartwatch. In *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 72–77. IEEE, 2018.

A Appendix: Details of Challenges in Generalized Gesture Recognition

Supervised learning methods: Gesture recognition works build specialized models for each new device (Huang et al., 2021; Alkiek et al., 2023; Wen et al., 2016a) or population (Khanna et al., 2024; 2023). TapNet (Huang et al., 2021) builds a gesture recognition model specifically for smartphones and does not work for different form devices like smartwatches or earbuds. Khanna et al. build a gesture recognition model trained for blind users (Khanna et al., 2024). This problem of lack of generalization can be solved if we have a lot of training data for gestures (hours of gesture data). However, obtaining training data at scale for gestures is challenging, costly, and time-consuming.

While recent *language-guided supervised frameworks* such as GOAT (Miao & Chen, 2024), IMUGPT 2.0 (Leng et al., 2024), and RAG-HAR (Sivaroopan et al., 2025) aim to address the lack of generalization by aligning IMU data with natural language, they do not scale to short-duration gestures. IMUGPT 2.0 acknowledges its inability to generate reliable 3D motion for subtle, fine-grained human movements. GOAT relies on 2-second windows to capture sufficient context for daily activities, which is far too long for millisecond-level gestures. Also, GOAT and RAG-HAR heavily depend on the pre-trained knowledge of the language model, potentially limiting their ability for activity descriptions unseen during pre-training.

Masking based self-supervised methods: Researchers in the HAR domain have made progress on this problem of generalization using self-supervised learning techniques (Xu et al., 2021; 2023a; Hong et al., 2024). These techniques use unlabeled human activity data to train a model and fine-tune it for different tasks, and cross-dataset for activity recognition.

Training a self-supervised HAR-like model for gesture recognition from scratch is not possible due to the limited availability of gesture data. To illustrate the data scarcity: the smallest publicly available HAR dataset contains 3.5 hours of data (Reyes-Ortiz et al., 2016), whereas the most extensive publicly available gesture dataset is only 0.8 hours (Porzi et al., 2013).

Our goal is to use human activity data for training gesture recognition since both rely on the same human movements, like lifting or rotating a limb. But as discussed in the previous section, using existing techniques (Xu et al., 2021; 2023a; Hong et al., 2024) for gesture recognition performs poorly. We find that this poor performance is largely because of how existing HAR-based self-supervised learning techniques perform masked modeling. Current works (Xu et al., 2021; 2023a; Hong et al., 2024) mask random segments anywhere in the signal, and the model learns to reconstruct them based on the surrounding temporal context. This approach works well for activities because information is distributed relatively evenly throughout repetitive motion patterns.

Figure 2 illustrates this. When masking 2 seconds in the middle of a walking sequence (a), the model observes before and after the masked region, identifies the activity context, and predicts continuation of the same pattern. The model learns what typically follows what across the sequence. However, for gestures (Figure 2(b)), random masking fails because it lacks the surrounding context to learn from. Unlike human activities, gestures are short and non-repetitive. In gestures, information is concentrated in a brief duration, while the rest of the signal is generic preparation or retraction. If a random mask covers these preparation or retraction areas, the model learns nothing meaningful because those parts carry no important information. Conversely, if the mask covers the gesture-defining region, the surrounding context is again not helpful for reconstruction. This is analogous to masking random characters in BERT instead of whole words; the masking is either irrelevant or destructive to capturing the meaningful unit.

Contrastive-learning based self-supervised methods: Beyond masked modeling, works such as ContrastSense (Dai et al., 2024) and ColloSSL (Jain et al., 2022) leverage contrastive learning to address domain shifts (e.g., cross-device placement) in HAR self-supervised training. These methods use positive and negative pairs to enable domain adaptations, such as cross-location and cross-dataset. In our evaluation, we pre-train ContrastSense on activity data and fine-tune it on gesture

data, but this approach performs poorly (§ 6.1). This is because such systems apply contrastive learning during the pre-training stage, where class labels are unavailable. Instead, they rely on heuristics of raw motion signals, such as timestamps or sample similarity, to generate positive and negative samples. For instance, ContrastSense (Dai et al., 2024) defines negative pairs using samples from distant timestamps (high likelihood of different activities) and positive pairs using adjacent samples (high likelihood of the same activity).

Instead, we find that applying contrastive objectives during the fine-tuning stage is more effective, as class labels are available at this stage (see §4). This enables the model to separate gestures based on semantic distinctions rather than relying solely on motion similarity.

B Appendix: Additional Details of Token-based Pre-training

This appendix expands on Stage 1 (§3). It gives an overview of the pre-training components and their motivation, details of nucleus detection and parameter selection, the focused masking tradeoff, the input encodings, and an extended discussion of the attention analysis.

B.1 Overview of the Pre-training Stage

GestureLens addresses the limitations of existing approaches through a two-stage framework. Stage 1 uses HAR data to pre-train a model that can be used for gesture recognition; Stage 2 fine-tunes the model for classifying gestures.

As discussed in §2, the problem with existing approaches is that they focus randomly across the motion signal, often missing information from the informative region of the motion. Traditional random masking used in HAR (Xu et al., 2021) does not work when applied to gesture recognition, as it misses the nucleus region. Instead, we develop token-based pre-training explicitly targeting the nucleus phase. We tokenize the motion data into three phases and focus on the most important token, the nucleus. *GestureLens* implements this focus and masks in the informative regions to ensure we reconstruct the nuances of a short motion.

Figure 4 shows the pipeline of the pre-training stage. We use unlabeled human activity data to train a self-supervised model that captures motion patterns. We identify the informative parts of the motion and mask within them. We support this learning with additional input encodings to obtain a robust model. The pre-training stage comprises the following components:

- **Nucleus Identification:** We implement a selection mechanism that automatically detects high-energy motion events in the unlabeled activity data (e.g., the swing phase between toe-off and heel strike in a walking sequence).
- **Focused Masking:** Unlike random masking as used in HAR approaches (Xu et al., 2021; 2023a), we place 80% of our masks *inside* the nucleus and only 20% outside. This forces the model to reconstruct the detailed internal dynamics of the motion event, rather than guessing based on the surrounding context.
- **Input Encodings:** We support this learning with two specific encodings: *Nucleus Encoding*, which explicitly marks the boundaries of the event, and *Significant Axis Encoding*, which identifies the sensor axis containing the relevant motion (e.g., the X-axis for a horizontal swipe), allowing the model to ignore noise.

The model thereby learns the patterns *within* a motion event rather than the context *around* it, and this representation transfers from continuous activities to isolated gestures.

B.2 Nucleus Detection: Parameter Selection and Robustness

A key challenge is that both the duration and magnitude of the nucleus vary across datasets and users. The nucleus is represented in terms of samples, which depend on the sampling frequency of the device, and its magnitude can vary arbitrarily based on how the gesture is performed and the sensor characteristics. To understand this variation, we analyze multiple gesture datasets and observe that while the temporal duration of the nucleus lies within a similar range (a short burst of motion), the number of samples representing it varies with sampling rate, and the signal magnitude can differ significantly across devices and users.

To make the detection consistent across datasets, we resample all signals to a common sampling frequency. This allows Δt in Eq. 1 to correspond to a fixed temporal window across datasets, rather than varying arbitrarily with sampling rate. We then choose Δt based on the temporal width of the nucleus, so the window captures the transition into and out of the core motion while smoothing out short fluctuations. We normalize the energy signal (scale it from 0 to 1) so that $motion_thresh$ is comparable across users and devices with different signal magnitudes. The threshold is selected

empirically using validation data from multiple datasets to consistently detect a continuous high-energy segment while excluding low-energy preparation and retraction phases. We use $\Delta t = 15$ samples (750 ms at 20 Hz) and $motion_thresh = 0.4$. This design captures changes in motion rather than absolute values, so detection depends on when the signal transitions, not how large it is.

Figure 7 shows examples of IMU signals from multiple datasets, including different devices and user populations. Each plot shows the motion signal over time, with the shaded region indicating the detected nucleus using our method. These examples are selected to highlight variation across datasets, including differences in sampling frequency, signal magnitude, and gesture execution. Despite these differences, using the same Δt and $motion_thresh$, the detected nucleus consistently aligns with the high-energy portion of the signal corresponding to the core motion. This demonstrates that the method identifies a continuous segment capturing the most informative part of the gesture, while excluding low-energy preparation and retraction phases, and that it is robust to variations across devices, users, sampling frequencies, and signal magnitudes.

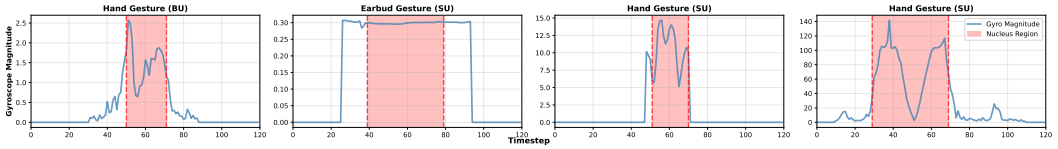

Figure 7: Nucleus detection across different datasets and devices. Each plot shows the IMU signal over time, with the shaded region indicating the detected nucleus. The method identifies a continuous high-energy segment corresponding to the core motion, while excluding low-energy preparation and retraction phases. Despite variations in signal magnitude and duration, the detected nucleus consistently aligns with the most informative portion of the gesture.

B.3 Focused Masking: Choice of Nucleus Masking Probability

Prior self-supervised HAR methods mask uniformly across the signal (Xu et al., 2021; 2023a), which is ineffective for short gestures where the informative content is concentrated in a brief segment. Our insight is that instead of masking uniformly, we should mask within the nucleus, as it contains the core motion of the gesture. We assign a higher masking probability (p_1) to samples inside the nucleus and a lower probability to samples outside, so that most masked samples correspond to the core motion.

This introduces a tradeoff. If p_1 is too low, most of the nucleus remains visible, and many masked samples fall in low-energy regions of the gesture (preparation or retraction phase). The reconstruction task becomes easy and the model does not learn the core motion patterns. If p_1 is too high, most of the nucleus is masked, leaving too little signal to capture the structure of the gesture, making the task overly difficult.

We select p_1 empirically through a parameter sweep on validation data. The chosen setting forces the model to reconstruct the core motion while retaining enough visible samples to preserve the structure of the gesture. We analyze this tradeoff in §6.2 (Table 5), showing that prioritizing masking within the nucleus leads to improved performance across different devices and populations.

Nucleus mask construction. We generate a binary mask $M_{nucleus}$ from the nucleus boundaries, where $M_{nucleus}(t) = 1$ if time t falls within a nucleus region and 0 otherwise. This single mask is used both for the nucleus encoding (§3.3) and to decide where masked spans are placed. Within the nucleus, masked positions are grouped into contiguous spans of 4–5 samples. In the pre-training loss, the set of masked indices M is therefore dominated by nucleus positions.

B.4 Input Encodings: Additional Details

Nucleus encoding. The nucleus encoding signals to the model that the marked region is where the motion patterns matter most. Time steps falling within the nucleus receive a distinct learnable vector compared to other time steps, which allows the self-attention mechanism to amplify focus on the nucleus token and attend less to the preparation and retraction phases. Without this guidance, attention disperses uniformly and misses the critical local patterns (Figure 6).

Significant axis encoding. We identify the discriminative axis within each token by analyzing rotational patterns across gyroscope dimensions during the nucleus. The axis with the highest cumulative rotation receives additional weight through a binary mask. This ensures tokens encode directional information alongside temporal patterns, creating representations sensitive to both motion

dynamics and spatial orientation. Without this component, the model learns directionally invariant features that cannot distinguish opposing movements such as swipe left and swipe right, or walking upstairs and downstairs.

Positional encoding. We augment these encodings with a positional encoding. Positional encodings have been used in IMU-based self-supervised methods to provide temporal ordering (Xu et al., 2021); a Transformer without positional encoding has no sense of temporal sequence. $E_{position}$ in Eq. 5 provides this temporal context. Together, the four input components guide self-attention away from uniform temporal processing toward token-focused attention.

B.5 Extended Attention Analysis

We validate that token-based pre-training creates focused attention through a comparative analysis with HAR-based pre-training (Xu et al., 2021). Figure 6 visualizes attention patterns across four motion types: smartwatch gestures, earbud interactions, blind user movements, and climbing activities. Without token-based pre-training, attention disperses uniformly across sequences, treating all temporal positions equally. With the focused strategy, attention concentrates into vertical bands aligned with the detected nucleus tokens: the model learns to focus where information exists. This targeted attention yields an 8.7% reduction in reconstruction MSE across test sequences.

More importantly, this validates the tokenization approach. The model no longer spends capacity learning from uninformative preparation and retraction regions, and instead develops representations centered on discriminative motion patterns. These token-aware representations transfer directly to fine-tuning for gesture classification: the model already knows where to look for critical information, enabling accurate classification even with minimal labeled examples of the target task.

C Appendix: Additional Details of the Text-guided Classifier

This appendix expands on Stage 2 (§4): the overall fine-tuning procedure, how text descriptions are constructed and used, and the loss components and their weighting.

C.1 Overview of the Fine-tuning Stage

We pass target data through the pre-trained model to obtain motion embeddings. The pre-trained encoder is frozen throughout Stage 2; its weights are not updated, so the same encoder is shared unchanged across all devices, populations, and tasks. We then use a small number of labeled examples of the target task to train a classifier that distinguishes between these embeddings. We build this text-guided classifier in two steps: first, we create text descriptions that capture the semantic properties of each gesture (§4); second, we use contrastive learning to organize the embedding space based on both motion patterns and semantic meanings (§4). The textual description is used only during training and is not needed at inference time. We use 10% of the total labeled data for fine-tuning, sampled in a stratified manner across classes. Architecture and optimization details for the classifier are given in Appendix D.

C.2 Text Descriptions

The text descriptions provide the semantic information used by the contrastive objective. Each class in the target dataset is assigned one description following the structured template in §4. These descriptions systematically capture movement direction (up, down, left, right, circular), motion type (swipe, rotation, shape-tracing), higher-level categorization (stationary, directional, rotational, shape, tap), and complexity (simple, complex).

This approach creates semantic relationships between gesture types that mirror human understanding. *Swipe up* and *swipe down* are recognized as related directional gestures, while *circle* and *figure-eight* share characteristics of shape-tracing gestures.

We pass these descriptions through a BERT language model. The output is a set of text embeddings that represent each gesture class in a higher-dimensional space. Using these embeddings, the model captures the semantic relationships between gestures: descriptions with similar meanings are mapped to points that are closer together in this space.

Cost and who provides the descriptions. Text description generation is a one-time preprocessing step. The developer manually provides one short description for each class in the target dataset using the structured template. Because there is only one description per class, the total amount of text is very small. Once the descriptions are processed through BERT, the resulting embeddings remain fixed. They are retrieved from storage during contrastive learning, so the model does not re-process text during training, and there is no additional computational overhead. No text is required at inference.

Robustness to description style. We evaluate sensitivity to the wording of descriptions, and the effect of replacing descriptions with shuffled, off-domain, one-hot, or random anchors, in §6.2.

C.3 Loss Components and Weighting

Classification loss. The classification loss is the standard cross-entropy loss, used in most classification problems. It establishes the basic decision boundaries on which the semantic and contrastive terms build.

Dynamic weighting. The final training objective is a weighted sum of the three components. The classification loss is the primary objective (implicitly weighted by 1), and the semantic and contrastive losses are added with gradually increasing weights w_s and w_c. Both weights start at 0.1 and increase during training to maximum values of 0.3 and 0.5, respectively.

This dynamic weighting allows the model to first establish basic decision boundaries through the classification loss, then progressively incorporate semantic relationships and contrastive separation as training progresses. The result is an embedding space in which gestures are organized not only by motion similarity but also by semantic meaning. Semantically related gestures are placed near one another, while semantically distinct gestures maintain clear separation even when their motion patterns overlap. This organization enables accurate classification with limited labeled examples, since the model can use the learned semantic relationships when direct training data for a class is scarce.

D Appendix: Implementation

In this section, we detail our implementation choices, model architecture, and training procedure.

D.1 Model Architecture

Our system consists of two main components: (1) pre-training a transformer-based model to understand the motion patterns, and (2) learning a text-guided classifier for gesture classification. The input to the model is IMU data, chunked into windows of 120 samples, resampled at 20Hz. If the signal length is $<$120, we append zeros at the start and end. This window length is an accepted practice in HAR models (Xu et al., 2021; 2023a).

We normalize the raw input accelerometer and gyroscope data before passing it to the transformer model. The transformer model processes the data through multiple input encodings to generate motion embeddings that capture the discriminative patterns of motion. The model learns to reconstruct masked segments of the input to ensure these motion embeddings contain meaningful representations. For downstream gesture classification, the motion embeddings are processed through a classifier that uses semantic information to recognize gestures across different form factors and user populations.

- **Pre-training Phase (Stage 1):** Our transformer model architecture is inspired by self-supervised works in HAR (Xu et al., 2021; 2023a). We re-imagine this pipeline by integrating domain-specific input encodings (§ 3.3) and replacing random masking with a focused masking strategy. The input to the model is 6-axis raw IMU data, projected through a linear layer to form the initial signal representation. This representation is then combined with positional, nucleus, and significant axis input encodings that guide attention to the discriminative motion patterns. Following the nucleus formulation in Eq. 1, we detect nucleus regions using a sliding-window-based energy change criterion, as described in §3.1. All datasets are resampled to a common frequency, so a fixed Δt corresponds to a consistent temporal window across datasets. We choose Δt based on the typical duration of the nucleus, so the window captures the transition into and out of the core motion while smoothing out short fluctuations. The threshold $motion_thresh$ is selected via grid search on validation data after normalizing the energy signal, choosing the value that consistently detects a continuous high-energy segment corresponding to the core motion while excluding low-energy preparation and retraction phases. During pre-training, we apply the focused masking strategy described in §3.1, which prioritizes masking within nucleus regions. Specifically, we choose values that ensure most masked samples lie within the nucleus while retaining sufficient visible samples to preserve the shape of the motion. The effect of this choice is analyzed in §6.2.

  The model used for pre-training is an encoder-decoder transformer block. The above-discussed encodings form the input for the encoder block used. The encoder transformer consists of 4 attention heads with a hidden dimension of 72 and a feed-forward dimension of 144. The model reconstructs masked motion segments using a lightweight decoder consisting of a linear projection

followed by GELU activation and layer normalization. The decoder maps the learned motion embeddings back to the original IMU signal space.

- **Fine-tuning Phase (Stage 2):** For gesture classification, we employ a transformer-based classifier that takes the motion embeddings from the pre-trained transformer model as input. The classifier consists of a transformer with 4 attention heads and 2 layers, followed by mean pooling over the sequence length. The transformer applies multi-head self-attention mechanisms and feed-forward processing to identify important patterns in the gesture sequence and generate two outputs: (1) a classification head that predicts gesture labels, and (2) a projection head implemented as a two-layer MLP that creates normalized motion embeddings for contrastive learning. During training, semantic embeddings derived from text descriptions of each gesture class are combined with motion embeddings to organize the embedding space based on both motion similarity and semantic meaning, as described in §4.

D.2 Training Procedure

We train our system in two distinct phases. During pretraining, we use unlabeled IMU data obtained from public human activity datasets (see § 5). Following the nucleus formulation in Eq. 1 and the focused masking strategy in §3.1, we set $\Delta t = 15$ samples (750ms at 20Hz) and $motion_thresh = 0.4$ for nucleus detection, selected via grid search. We apply the focused masking strategy with $p_1 = 0.8$ in Eq. 2, selected based on validation performance (Section 5). Approximately 15% of the sequence values are masked, with the majority of masking applied within nucleus regions. The model learns to reconstruct these masked values. We use the Adam optimizer with a learning rate of 1e-3 and train for 3200 epochs, with early stopping, batch size of 128. The loss function is MSE between the reconstructed and original values. We use 80% of the activity data for training this model, and use the remaining 20% for validation and test sets.

For the gesture recognition task, we implement a combined loss function comprising classification, text-guided, and contrastive loss. We employ dynamic weights for the text-guided and contrastive losses, which increase gradually during training from 0.1 to their maximum values of 0.3 and 0.5, respectively. This allows the model to first focus on classification before incorporating text-guided and contrastive objectives. We train the classifier for 200 epochs with a batch size of 64 using the Adam optimizer, using early stopping. Hyperparameters are selected via grid search over these parameters: learning rate $\{1e{-}4, 3e{-}4\}$, weight decay $\{1e{-}4, 5e{-}4\}$, contrastive loss weight $\lambda_2 \in \{0.001, 0.005\}$, and gradient clipping norm $\{0.5, 1.0\}$. We select the best configuration based on validation accuracy and F1 score. For training, we use 10% of the labeled data, sampled in a stratified manner to ensure balanced representation across all classes.

E Appendix: Evaluation Setup

E.1 Additional Dataset Details

Gesture Recognition Datasets.

- *Sony Watch Gesture (alias SU Watch Gesture):* This dataset (Porzi et al., 2013) is publicly available and includes accelerometer data from a Sony SmartWatch worn on the user's right wrist. Eight sighted users performed twenty repetitions of twenty different gestures, totaling 3200 sequences. Gestures were manually segmented by users tapping the smartwatch screen at the beginning and end of each repetition. Data were collected at 10 Hz. Using this dataset allows us to evaluate the system's ability to accurately recognize a high number of distinct gesture classes (20), showing our text-guided classifier's capacity to differentiate subtle variations in movement.
- *UMAHand:* We use the UMAHand dataset (Casilari et al., 2024), which includes 25 volunteers (16 female, 9 male) aged 18 to 56. Data were collected at 100 Hz using a Shimmer 3 sensor node worn on the participant's dominant wrist. From the full set of 29 activities, we select four classes that can be characterized as hand gestures: clapping hands, waving, sweeping, and screwing.
- *UTD-MHAD:* We use the UTD-MHAD dataset (Chen et al., 2015), which features 8 participants (4 female, 4 male) performing actions 4 times each. The dataset contains 3-axis accelerometer and gyroscope data sampled at 50 Hz from the sensor worn on the user's right wrist. Similar to UMAHand, we select 8 hand gesture classes from the 27 available actions, excluding full-body activities.
- *Mechanomyographic (MMG) Gesture:* We evaluate the system using the MMG dataset (Abbas et al., 2025), which includes 43 participants (60% male, 40% female) ranging in age from 18 to 69. Participants performed 50 repetitions of 11 different hand gestures (e.g., finger snapping, fist

making, thumb tapping). Data was captured at 200 Hz using a wearable system placed just below the wrist strap. We select this dataset to evaluate *GestureLens* on a larger and demographically diverse user pool with fine-grained hand gestures.

- *Sighted User (SU) Earbud Gesture:* We collected earbud data from 9 sighted participants (3 male, 6 female) between ages 22-29, performing seven gestures on the earbud: tap, double tap, long press, swipe up, swipe down, rotate on the surface, and tap on the lower end. Each gesture was repeated five times. Each gesture lasted about 0.8 seconds. Similar earable gestures have previously been explored for interaction (Alkiek et al., 2022; Rateau et al., 2022). We collected this data using (Kawsar et al., 2018), an accelerometer, and a gyroscope sampled at 20 Hz. We use this dataset to evaluate the performance of *GestureLens* for different form factors, specifically for earworn devices. This dataset, while featuring fewer gestures (7 classes), is noteworthy for its relatively small number of samples (315), showing the system's performance with limited training data.
- *Blind User (BU) Watch Gesture:* We collected this dataset from 10 blind participants (5 male, 5 female) between ages 38-64, with 8 participants blind since birth and 2 participants blind since ages 3 and 11, respectively. Each participant performed 12 different gestures 10 times each while wearing a Fossil Gen 5 smartwatch, and sampled the data at 100 Hz for the accelerometer and gyroscope. These gestures included five forearm directional gestures, five compound gestures, and two shape gestures. Figure 10 shows the 12 gestures used. We chose these gestures because they have previously been explored for accessible interactions (Khanna et al., 2023; 2024). The duration of the gesture averages 1.3 seconds, which is longer than gestures performed by sighted users. This dataset allows us to evaluate cross-population performance, as the high inter-user variance among blind users demonstrates the system's strong generalization capabilities.

Human Activity Recognition Datasets. We also evaluate *GestureLens* on four public HAR datasets used in prior work (Xu et al., 2021; 2023a; Hong et al., 2024). *HHAR* (Stisen et al., 2015) contains accelerometer and gyroscope data from 9 users performing 6 activities with 6 phone types worn at the waist, sampled at 100-200 Hz. *UCI* (Reyes-Ortiz et al., 2016) includes 30 volunteers performing 6 basic activities with a waist-mounted smartphone at 50 Hz. *MotionSense* (Malekzadeh et al., 2019) comprises accelerometer and gyroscope data from 24 diverse participants performing 6 activities with an iPhone in their front pocket at 50 Hz. *Shoaib* (Shoaib et al., 2014) collected data from 10 participants performing 7 activities with smartphones at five body positions, recording accelerometer, gyroscope, and magnetometer readings at 50 Hz.

We use the unlabeled *HHAR* dataset for pre-training our model; 80% of the data goes into training, 10% into validation, and the remaining 10% for testing. We use *UCI*, *MotionSense*, and *Shoaib* datasets in a cross-dataset manner (train on one and test on another) to evaluate *GestureLens* for the human activity recognition task.

E.2 Baselines

We compare *GestureLens* against a wide set of baselines spanning supervised, self-supervised, and contrastive learning paradigms. We include representative self-supervised and contrastive models such as LIMU-BERT (Xu et al., 2021), UniHAR (Xu et al., 2023a), ContrastSense (Dai et al., 2024), ColloSSL (Jain et al., 2022), and Cosmo (Ouyang et al., 2022). We further include recent foundation and large-scale models, including Qiu et al. (Qiu et al., 2025) and Yuan et al. (Yuan et al., 2024), along with text-guided approaches such as IMUGPT 2.0 (Leng et al., 2024). In addition, we include specialized gesture recognition systems designed for specific form factors or user populations, such as Serendipity (Wen et al., 2016b), EarBender (Alkiek et al., 2023), and Khanna et al. (Khanna et al., 2024). The baselines are selected based on prior performance, relevance to the task, and code availability (Table 1). For all baselines, we perform hyperparameter tuning over epochs, number of layers, and batch size to ensure a fair comparison.

E.2.1 Supervised and Self-supervised baselines

- **DeepSense** (Yao et al., 2017) (supervised). It is a gesture recognition model based on supervised learning. It applies the Fourier Transform to the raw IMU data and feeds the frequency domain features (i.e., magnitude and phase pairs) to the neural network. It adopts a deep learning structure to fuse data from multiple sensors and accordingly extract temporal features.
- **LIMU-BERT** (Xu et al., 2021) (self-supervised). LIMU-BERT is an activity recognition model that learns representations from unlabeled IMU data using self-supervised training inspired by the BERT language model. It extracts generalized rather than task-specific features through masked

sensor prediction. The learned representations can be used to train downstream task-specific models with limited labeled data (as we do for gesture recognition).

- **ColloSSL** (Jain et al., 2022) (self-supervised). ColloSSL is a self-supervised learning technique that leverages unlabeled data from multiple devices worn by the same user. It treats different sensor streams for the same activity as natural transformations of one another to generate a signal for representation learning.
- **Cosmo** (Ouyang et al., 2022) (self-supervised). Cosmo is a contrastive fusion learning system designed for multimodal human activity recognition with limited labeled data. It employs a two-stage strategy that uses unlabeled data on the cloud to pre-train a fusion model and then fine-tunes it on the edge using limited labels. Although Cosmo is designed for multimodal fusion (e.g., IMU and video), we evaluate it in our single-modality context (IMU) to see how its contrastive fusion architecture handles label scarcity for gesture recognition.
- **UniHAR** (Xu et al., 2023a) (self-supervised). UniHAR is an activity recognition model that addresses data heterogeneity in different datasets by augmenting IMU data based on the physics of the sensing process. It integrates federated learning and adversarial training to improve generalization across users. For evaluation, we use the HHAR dataset for fine-tuning UniHAR and deploy it in a data-centralized manner.
- **ContrastSense** (Dai et al., 2024) (self-supervised). ContrastSense is closest to *GestureLens* and designed for gesture and activity recognition. It tackles domain shifts and label scarcity in wearable sensing through domain-invariant contrastive learning. It leverages unlabeled data with contrastive learning in the pretraining step. We use token-based pre-training to extract embeddings and use a text-guided classifier for gesture recognition.
- **Yuan et al.** (Yuan et al., 2024) (self-supervised). This model is a large-scale foundation model for activity recognition trained on 700,000 person-days of accelerometer data from the UK Biobank. It focuses on multi-task self-supervision to learn generalizable features that can be applied to diverse devices. We use this as a baseline to compare our system against a model trained on a massive, population-scale dataset, testing how well it transfers to fine-grained gesture recognition.
- **Qiu at al.** (Qiu et al., 2025) (self-supervised). This work proposes a foundation model framework for human activity recognition (HAR) that can be customized for specific wearable devices. We utilize its two-stage training model, which first extracts cross-domain sensor patterns using self-supervised learning and then aligns these with semantic contexts.
- **Bio-PM** (Tarale et al., 2026) (self-supervised). Bio-PM tokenizes continuous IMU signals into movement units based on acceleration zero-crossings and uses these units as tokens for masked pre-training. We evaluate Bio-PM to examine whether this movement-level tokenization strategy, designed for continuous human activity, transfers to short-duration gesture recognition.

For the abovementioned baselines, we use their pretrained encoders to generate embeddings and then fine-tune their classifiers using a stratified 10% subset of the labeled data, with hyperparameter tuning.This ensures that all models are evaluated under the exact same data conditions as *GestureLens*.

E.2.2 GESTURE RECOGNITION FOR SPECIALIZED FORM FACTORS AND POPULATIONS:

We also compare our generalized gesture recognition system with specialized algorithms:

- **EarBender** (Alkiek et al., 2023) (supervised). EarBender is a gesture recognition system for touch-based hand-to-ear gestures. It uses IMUs in commodity earables to recognize gestures. The system detects different actions and employs a 1D-CNN for classification and data augmentation. EarBender is designed to be user-invariant. We implemented EarBender's architecture and preprocessing techniques, adapting it to the sighted user's earbud gesture dataset.
- **Serendipity** (Wen et al., 2016a) (supervised). This is a smartwatch-based hand gesture recognition framework. This work trains a personalized SVM classifier. We trained a personalized model for each user in the sighted user hand gesture dataset using the features described in the paper. The features are extracted from the time-series data obtained from both the accelerometer and the gyroscope, and take into consideration the orientation in which the gesture was performed.
- **Hand Gesture Recognition for Blind Users** (Khanna et al., 2024) (supervised). This work presents a hand gesture recognition system for blind users by 3D gesture trajectory tracking. It

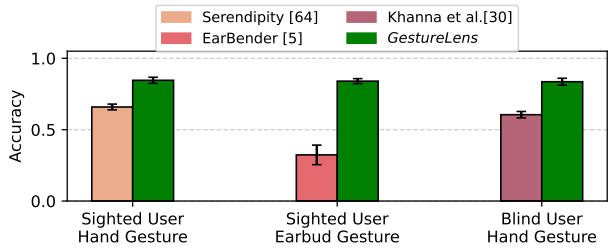

Figure 8: Comparison of *GestureLens* with dataset-specific gesture recognition methods for SU-HG, SU-EG, and BU-HG. In addition to high accuracy for multiple datasets, *GestureLens* also outperforms specialized approaches.

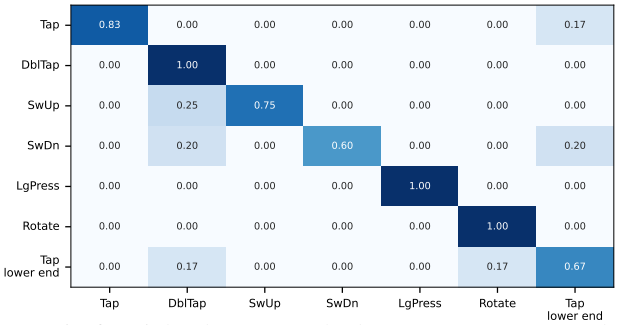

Figure 9: Confusion matrix for sighted users' earbud gestures. Pronounced gestures like tap and double have slightly higher performance than more subtle gestures like swipe up and swipe down.

employs an ensemble classifier that combines a multi-view CNN model with geometric properties analysis using only gyroscope data. We implemented their ensemble classifier architecture, adapting it to our blind user hand gesture dataset.

For these baselines, we used 10% of the labeled data for training, sampled in a stratified manner to ensure a balanced representation across all classes: 12 blind user hand gestures, 7 sighted user earbud gestures, and 20 sighted user hand gestures.

F Appendix: Additional Results

F.1 Comparing with Specialized Gesture Recognition Algorithms

Beyond reporting overall accuracy, we evaluate *GestureLens* under more targeted settings, performance against specialized gesture recognition systems, the effect of varying the number of gesture classes (Section F.3) and data size (Section F.4), as well as the impact of removing individual system components (Section 6.2). We perform these evaluations focusing on a subset of datasets representing cross-user (BU hand gestures), cross-form-factor (SU earbud gestures), and cross-population (SU hand gestures) settings. We compare against baselines spanning supervised, self-supervised, and contrastive learning approaches, including DeepSense, LIMU-BERT, UniHAR, ContrastSense, and Cosmo.

After evaluating against generalizable recognition models (Section 6.1), we compare *GestureLens* with methods specifically designed for gesture recognition on individual datasets: Serendipity (Wen et al., 2016b) for sighted hand gestures, EarBender (Alkiek et al., 2023) for ear-based gestures, and Khanna et al. (Khanna et al., 2024) for blind user hand gestures. This allows us to assess how *GestureLens* performs relative to approaches specifically designed for certain form factors and populations, under limited labeled data. Figure 8 shows that none of the dataset-specific models achieve the accuracy of *GestureLens*. As before, this is because the baseline models rely on relatively large labeled datasets for effective training. In contrast, *GestureLens* leverages unlabeled data through token-based pretraining to learn meaningful human motion representations, enabling the text-guided classifier to build accurate classifiers from only a few labeled examples per class. This is further reflected in the confusion matrices (Figures 11 and 10), where *GestureLens* exhibits reduced confusion between gestures with similar motion. For instance, Figure 11 shows clear separation between "tap" and "double tap", while Figure 10 shows high discrimination between "forearm up" and "forearm down".

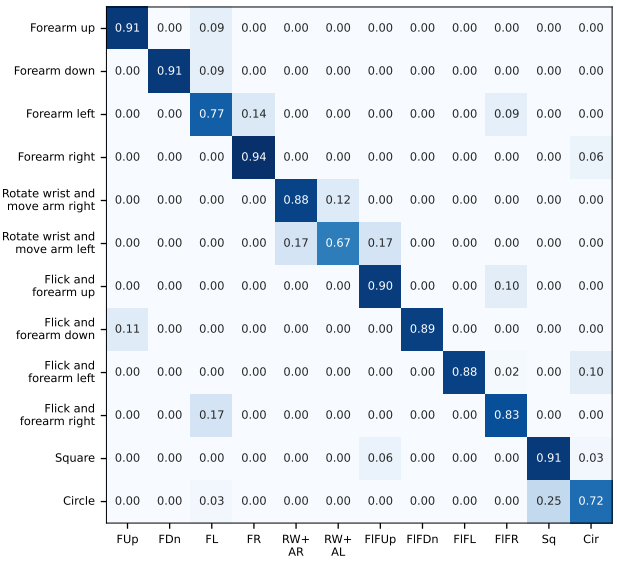

Figure 10: Confusion matrix for blind users' hand gestures. Simple directional movements achieve >90% accuracy, while compound gestures show minimal confusion between similar movements.

F.2 Confusion Matrix

Figure 10 shows the confusion matrix for the blind user hand gesture data. This matrix allows us to analyze the classes of gestures for which *GestureLens* achieves high vs low accuracy. *GestureLens* is able to detect simple directional gestures with high accuracy: for instance, the accuracy for forearm up, forearm down, and forearm right are 91%, 91%, and 94%, respectively. However, more complex combinations of movements show poorer accuracy. For example, "rotate wrist and move arm left" achieves 67% accuracy, with some confusion with "rotate wrist and move arm right" (17%) and "rotate wrist and move arm down" (17%). Similarly, circle gestures achieve 72% accuracy, with some confusion with square gestures (25%). We observe a similar trend for sighted users' earbud gestures (Figure 11) where more pronounced gestures like tap and double tap have higher accuracy than finer gestures of swipe up and swipe down. However, the average accuracy remains 85% across the sighted-user hand, sighted-user earbud, and blind-user hand gesture datasets. (Figure 12).

In the future, we will develop techniques to disambiguate such similar and complex human motion.



Figure 11: Confusion matrix for sighted users' earbud gestures. Pronounced gestures like tap and double have slightly higher performance than more subtle gestures like swipe up and swipe down.

F.3 Effect of Increasing Gesture Classes

As interaction systems evolve, users often require larger gesture vocabularies—whether to support more complex interactions as they become familiar with a system or to provide comprehensive accessibility options for diverse user needs. We evaluate *GestureLens*'s ability to handle these requirements by increasing the number of classes and reporting classification accuracy in Figure 13.



Figure 12: Confusion matrix for sighted users' hand gestures.

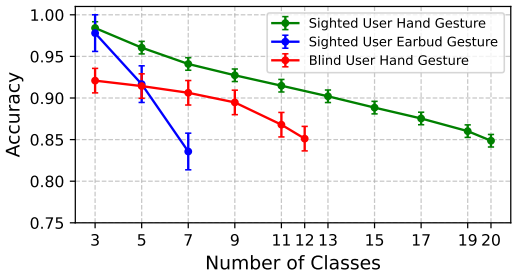

Figure 13: Performance of *GestureLens* as the number of gesture classes increase. *GestureLens*'s performance gracefully degrades as the number of classes increases, but the overall accuracy is still high even when disambiguating across a large number of classes.

For *SU-Hand gestures* (20 classes), *GestureLens* achieves above 85% accuracy even at 20 classes, with a gradual decrease from approximately 98% at 3 classes. This slow degradation curve suggests scalability for typical hand gesture interactions. Similarly, for *SU-Earbud gestures* (7 classes), *GestureLens* maintains high accuracy (above 95% for 3-5 classes), though with a more pronounced drop to about 84% at 7 classes. This steeper decline reflects the inherent challenge of distinguishing between subtle earbud gestures as the discrimination task becomes more complex. For *BU-Hand gestures* (12 classes), *GestureLens* achieves over 90% accuracy at 3 classes and maintains performance above 85% up to 12 classes. These results show *GestureLens*'s scalability with an increasing number of gesture classes across different form factors and user populations.

F.4 Performance with Limited Training Data

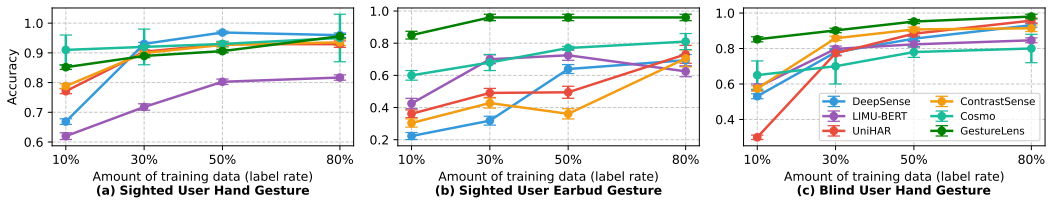

Figure 14: Effect of different fractions of labeled training data used to train the gesture classifier. *GestureLens* maintains consistently high performance even with limited training data, while baseline models show greater dependence on training data quantity.

We investigate the performance of *GestureLens* with varying amounts of labeled training data, ranging from 10% to 80% of the available data. Recall that the default is to use 10% labeled data. This scenario reflects real-world conditions where collecting large labeled datasets can be challenging, especially for novel gesture types or diverse user populations. For SU-Hand gesture recognition (Figure 14a), *GestureLens* achieves high accuracy (85%) even with only 10% labeled data and

maintains consistent performance as the label rate increases. It's worth noting that because 10% of SU-Hand gesture data is approximately 10× and 2.6× larger than other datasets, there is still sufficient data for other baselines (91% for Cosmo (Ouyang et al., 2022)) to perform reasonably well with hand gestures, which might not be the case for newer classes of gestures like earbud movements or data from accessory populations.

For *SU-Earbud gesture recognition* (Figure 14(b)), *GestureLens* maintains >80% accuracy for all label rates, showing its ability to extract meaningful features even from minimal data. The baselines exhibit greater performance variability, with DeepSense, in particular, struggling at low label rates (35% accuracy at a 30% label rate). This highlights one of the key differentiating factors of our system from baselines: utilizing supportive embeddings and text-guided contrastive learning to learn and classify meaningful representations. Similarly, for *BU-Hand gesture recognition* (Figure 14c), *GestureLens* performs better across all label rates. These results demonstrate *GestureLens*'s capability to work with limited labeled data, which is valuable in contexts where labeled data collection is challenging or expensive.

F.5 Ablation Continued...

Effect of nucleus masking ratio. We next analyze the role of the masking strategy used during pretraining. Table 5 evaluates different masking probabilities within the nucleus, i.e., the probability that a masked sample falls within the nucleus region. At lower masking probabilities (50%–70%), the probability of masking inside the nucleus is low, so most of the nucleus remains visible. In many cases, the masked samples fall outside the nucleus, and the model ends up reconstructing low-energy regions instead of the actual motion. This makes the task easy and does not force the model to learn the core motion patterns.

Masking %	Hand Gesture (SU)		Earbud Gesture (SU)		Hand Gesture (BU)	
	Acc.	F1	Acc.	F1	Acc.	F1
50%	0.84	0.84	0.79	0.77	0.77	0.75
70%	0.76	0.76	0.72	0.71	0.79	0.78
80%	**0.86**	**0.85**	**0.85**	**0.84**	**0.85**	**0.85**
90%	0.79	0.79	0.75	0.73	0.69	0.66

Table 5: Ablation of masking probability within the nucleus. Performance peaks at 80%, indicating a balance between reconstruction difficulty and available structure.

At higher masking probabilities (90%), almost the entire nucleus is masked. In this case, the model has very little access to the actual motion, and the remaining signal is not sufficient to capture the structure of the gesture. The reconstruction task becomes too hard, and the model loses the signal. A masking probability of 80% within the nucleus provides a balance between these two extremes. It hides most of the core motion, forcing the model to learn meaningful patterns, while still keeping enough of the nucleus visible to preserve the structure and shape of the gesture.

Effect of varying textual descriptors We next examine whether the gains from the text-guided classifier arise from the semantic information in the gesture descriptions, rather than simply from introducing an additional classification objective. We progressively remove or corrupt the semantic information while keeping the overall classification framework unchanged. Shuffling descriptions across classes, replacing them with off-domain descriptions, and using one-hot or random anchors all reduce performance relative to the full text-guided model, with the largest degradation generally observed when semantic information is removed entirely. Similarly, excluding real same-class pairs or removing the semantic loss leads to lower performance across the evaluated datasets. These results (shown in Table 6) indicate that the classifier benefits from meaningful correspondence between motion representations and their textual descriptions, rather than from the presence of an auxiliary text component alone.

F.6 Pretraining vs. Classifier Impact

To disentangle the contributions of motion representation learning and the text-guided classifier, we replace the *GestureLens* encoder with pretrained representations from existing HAR methods while keeping the downstream classification framework fixed. We evaluate each encoder with both a standard cross-entropy (CE) classifier and our text-guided objective (Table 7. This allows us to ask whether the gains from text guidance transfer directly to representations learned by existing methods,

Text condition	Hand Gesture(SU)		Earbud Gesture(SU)		Hand Gesture(BU)	
	Acc.	F1	Acc.	F1	Acc.	F1
Shuffled descriptions	0.688	0.679	0.72	0.71	0.775	0.774
Off-domain descriptions	0.699	0.691	0.74	0.73	0.783	0.775
One-hot anchors	0.696	0.689	0.73	0.72	0.787	0.789
Random anchors	0.670	0.663	0.69	0.68	0.771	0.771
Real, same-class pairs excluded	0.673	0.664	0.70	0.69	0.780	0.773
No semantic loss	0.685	0.679	0.71	0.70	0.772	0.770
GestureLens (full system)	**0.86**	**0.85**	**0.85**	**0.84**	**0.85**	**0.85**

Table 6: SHUFFLED permutes real descriptions across classes so none keeps its own; OFF-DOMAIN substitutes unrelated text in the same template; ONE-HOT and RANDOM strip semantic content entirely while keeping a per-class anchor; NO SEMANTIC LOSS removes the term. *GestureLens* (full system) reports the complete end-to-end model with real text descriptions and full text-guided classifier.

Encoder	Classifier	Hand Gesture(SU)		Hand Gesture(BU)	
		Acc.	F1	Acc.	F1
UniHAR Xu et al. (2023b)	CE only	0.148 ± 0.005	0.167 ± 0.011	0.216 ± 0.027	0.214 ± 0.019
	Text-guided	0.142 ± 0.003	0.166 ± 0.004	0.211 ± 0.016	0.212 ± 0.009
ContrastSense Dai et al. (2024)	CE only	0.624 ± 0.036	0.617 ± 0.032	0.671 ± 0.039	0.664 ± 0.037
	Text-guided	0.643 ± 0.031	0.639 ± 0.024	0.680 ± 0.048	0.672 ± 0.048
Yuan et al. Yuan et al. (2024)	CE only	0.562 ± 0.020	0.544 ± 0.034	0.385 ± 0.037	0.373 ± 0.047
	Text-guided	0.573 ± 0.034	0.554 ± 0.051	0.365 ± 0.081	0.346 ± 0.092
GestureLens†	CE only	0.79	0.78	0.73	0.73
	Text-guided	0.86	0.85	0.85	0.85

Table 7: Impact of encoder pretraining and classifier design. CE ONLY uses the standard cross-entropy objective, while TEXT-GUIDED additionally incorporates the semantic and contrastive objectives. Results are reported as Accuracy/F1-score.

and whether the combination of our pretraining and classifier is necessary for the improvements observed by *GestureLens*.

For UniHAR Xu et al. (2023a), adding the text-guided objective does not improve performance, with accuracy/F1 changing from 0.148/0.167 to 0.142/0.166 on SU hand gestures and from 0.216/0.214 to 0.211/0.212 on BU hand gestures. ContrastSense Dai et al. (2024) shows a modest improvement over text guidance, increasing from 0.624/0.617 to 0.643/0.639 on SU hand gestures and from 0.671/0.664 to 0.680/0.672 on BU hand gestures. For Yuan et al. Yuan et al. (2024), text guidance produces a small improvement on SU hand gestures (0.562/0.544 to 0.573/0.554), but slightly reduces performance on BU hand gestures (0.385/0.373 to 0.365/0.346). Thus, text guidance alone does not consistently produce large gains when applied to representations learned by existing HAR encoders.

In contrast, the complete *GestureLens* model benefits substantially from the text-guided objective. On SU hand gestures, text guidance improves performance from 0.79/0.78 with CE only to 0.86/0.85, while on BU hand gestures it improves performance from 0.73/0.73 to 0.85/0.85. The substantially larger gains observed when the text-guided classifier is paired with the *GestureLens* encoder suggest that the two components are complementary: the proposed pretraining produces motion representations that are particularly amenable to semantic alignment, while the text-guided classifier exploits this structure to improve gesture separability. These results indicate that the performance gains of *GestureLens* cannot be attributed to the classifier alone, but instead arise from the combination of the learned motion representation and semantic guidance.

F.7 BEYOND GESTURE RECOGNITION: CROSS-TASK HAR

Even though *GestureLens* is designed for gesture recognition, we show that it works well for other related tasks: human activity recognition (HAR). To evaluate *GestureLens* for HAR, we pre-train the

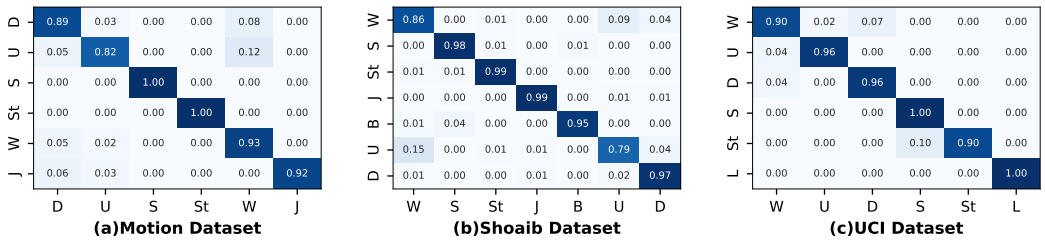

Figure 15: Performance of *GestureLens* for cross-task from gesture recognition to human activity recognition. The model is pre-trained on the unlabeled HHAR (Stisen et al., 2015) dataset and fine-tuned using 10% labeled data from the test set. The average accuracy is 93% across three test HAR datasets: Shoaib, MotionSense, and UCI.

motion model using the HHAR dataset (Stisen et al., 2015) (without any labels) and then test it on the other three HAR datasets after fine-tuning the model with 10% labeled data from the test set. This is the cross-dataset split for HAR performed in previous works (Xu et al., 2023a; Hong et al., 2024). Figure 15 shows the confusion matrix when *GestureLens* is trained on HHAR and tested over the Shoaib (Shoaib et al., 2014), MotionSense (Malekzadeh et al., 2019), and UCI (Reyes-Ortiz et al., 2016). Across these datasets, *GestureLens* achieves an average accuracy of 93%, even when trained on only 10% labeled data. This is very close to the state-of-the-art model UniHar (Xu et al., 2023a), which is specifically designed for HAR (94% accuracy). UniHar is specifically designed to work well for cross-dataset activity recognition. In contrast, LIMU-BERT (Xu et al., 2021), which is not designed for this cross-dataset scenario, only achieves an accuracy of 40% when trained on 10% labeled data. The result shows that the token-based pre-training we design in *GestureLens* is able to capture human motion well enough to recognize both short-duration gestures and longer activities.

The practical implication is that developers could potentially leverage *GestureLens* as a foundation for both gesture interfaces and activity monitoring applications—from smart home systems that recognize fine-grained gestures to healthcare applications that track coarse-grained activities, all with minimal dataset-specific training.

F.8 Smartphone Implementation

We implement the entire *GestureLens* pipeline on an Android phone and smartwatch. For mobile deployment, we transform our two-stage architecture into a unified TorchScript model, which provides an optimized, serialized representation that can execute efficiently on mobile devices. We utilize PyTorch Mobile's (PyT) 8-bit dynamic quantization, which converts 32-bit floating-point weights to 8-bit integers during inference. This quantization process is applied post-training and optimizes transformer components without requiring model retraining. The mobile implementation preserves the motion signature detection and significant axis identification algorithms, executing them efficiently in native Java code before model inference. The quantized model size was $863.16kB$. The end-to-end latency for gesture recognition across 10 trials on a Samsung Galaxy S20 phone was 66.3 msec, well below the latency that can be perceived by humans (Laming, 1968). On average, it took 46 msec to stream gesture data via Bluetooth from the smartwatch to the phone, and it took 20.3 msec to run the model inference on the phone. Together, these results show that *GestureLens* not only achieves high accuracy for multiple user populations and devices with minimal labeled data, but also supports end-to-end, real-time gesture recognition on commodity smartphones.

G Appendix 6: Additional Related Work

Gesture recognition systems have evolved from early approaches with hand-crafted features to advanced machine learning techniques extracting higher-dimensional features. Current research in this field primarily takes two directions: traditional recognition methods requiring large labeled datasets, or self-supervised learning approaches that utilize unlabeled data to learn representations, thereby reducing dependency on extensive labeled datasets.

Self-supervised Learning for IMU-based Sensing: LIMU-BERT (Xu et al., 2021) and its follow-up UniHar (Xu et al., 2023a) are amongst the first papers that use self-supervised learning using unlabeled HAR data; but the focus in both papers is human activity recognition (HAR). We already discussed and demonstrated the shortcomings of these models for the gesture recognition task. CrossHAR (Hong et al., 2024) addresses data heterogeneity challenges through physics-based data

augmentation and hierarchical contrastive sensor data pretraining, improving cross-dataset generalization. However, CrossHAR has poorer performance compared to UniHar for HAR, and the source code is unavailable, so we omit it from our baseline. The most recent work in this series, ContrastSense (Dai et al., 2024), is closely related to *GestureLens*. It proposes domain-invariant contrastive learning by leveraging domain labels to minimize discrepancies across users and devices and targets both activity recognition and gesture recognition. In contrast to *GestureLens*, ContrastSense utilizes sample similarity to create negative pairs with the expectation that they are distinct. Since various gestures share comparable motion signals, the system identifies these similar motions as negative pairs and attempts to separate them within the feature space. Beyond self-supervised learning, transfer learning and domain adaptation techniques have been explored. While some transfer learning systems for gesture recognition require extensive base model training data (Qi et al., 2023) or report low accuracy (Kang et al., 2021), domain adaptation specifically targets generalization challenges across varying IMU data conditions. Notable works include Soleimani and Nazerfard's (Soleimani & Nazerfard, 2021) use of GANs for cross-subject transfer learning in HAR, Sanabria and Ye's (Sanabria et al., 2021) unsupervised adaptation for heterogeneous datasets, and Hu et al.'s (Hu et al., 2023) SWL-Adapt for cross-user variability in wearable HAR. Though primarily focused on activity recognition, these methods' strategies for robust generalization are highly relevant to gesture recognition across diverse populations and form factors. Beyond single-device SSL, contrastive and multimodal strategies have been explored. ColloSSL (Jain et al., 2022) leverages collaborative learning across multiple devices, yet this introduces a dependency on multi-sensor environments that is impractical for the single-device focus of *GestureLens*. Cosmo (Ouyang et al., 2022) addresses label scarcity through multimodal fusion (e.g., IMU and video). While effective, such architectures rely on power-intensive or privacy-invasive secondary modalities, whereas *GestureLens* achieves robustness using only a single, modality-agnostic IMU stream. Qiu et al. (Qiu et al., 2025) aim to handle heterogeneity across different devices, yet they remain primarily evaluated on coarse-grained activity recognition rather than a cross-population use case. Similarly, Yuan et al. (Yuan et al., 2024) leverage a population-scale dataset of 700,000 person-days for pretraining, but its optimization for general daily activities does not necessarily translate to discrete, short-duration gestures on various populations and form factors.

Gesture Recognition for blind users: Gestures offer significant benefits for accessibility, especially for visually impaired users (Leporini et al., 2012; Xu et al., 2019; Azenkot & Lee, 2013; Kane et al., 2011). Research shows gestures performed by blind users differ substantially from sighted users, with higher inter-user variance, possibly due to lack of visual feedback, rendering recent recognition works impractical without large datasets (Khanna et al., 2023). While Modzelewski et al.(Modzelewski & Kaiser, 2012) demonstrate gesture recognition for blind users, their approach requires custom hardware. AccessWear(Khanna et al., 2023) works with commercial smartwatches but supports only 5 gestures. Khanna et al. (Khanna et al., 2024) expand the gesture set by designing a gesture recognition algorithm that uses 3D representation. Our evaluations show that this technique does not work well when only a small amount of labeled data is available.

Deep Learning using large datasets: Several gesture recognition algorithms train deep learning models such as CNNs and RNNs (Mohanty et al., 2017; Chung et al., 2019; Al-Hammadi et al., 2019; Prakash et al., 2020) using labeled data. These systems extract temporal and spatial features from IMU streams, showing promising results for smartphones, smartwatches, and earables (Shin & Sung, 2016; Zhu et al., 2018). Other systems have utilized commercial wearable devices for gesture recognition. TapNet (Huang et al., 2021), Serendipity (Wen et al., 2016b), and EarBender (Alkiek et al., 2023) are designed to recognize smartphone, smartwatch, and earbud gestures, respectively, using IMU. Previous works have shown that these existing approaches, designed primarily for sighted user gestures, do not work well when applied to blind user gestures (Khanna et al., 2024). Our evaluation of Serendipity (Wen et al., 2016b), and EarBender (Alkiek et al., 2023) shows that these techniques do not work well even for the scenario they were designed for, with limited data. Recent advancements leverage natural language processing and generative AI, such as IMUGPT 2.0 (Leng et al., 2024) and GOAT (Miao & Chen, 2024), to generate synthetic IMU data or align activity representations for improved HAR generalization with minimal data. While promising for data-scarce HAR, they can not be directly applied to gesture recognition, with its shorter durations and subtle variations, which warrants further investigation.

Alternative sensor modalities for gesture recognition There has been considerable work on using alternate sensor modalities, beyond IMUs, for gesture recognition. For instance, systems have used EMG (Chen et al., 2020; Wei et al., 2019), microphones (Das et al., 2017; Watanabe & Terada, 2018; Wang et al., 2019), and cameras (Oudah et al., 2020; Murthy & Jadon, 2009; Shahi et al., 2024)

amongst other sensors. We start with IMU-based gesture recognition because IMUs are available in most commercial devices. We leave the exploration of alternate sensor modalities for future work.

In summary, existing gesture recognition approaches require substantial labeled data or struggle with cross-user and cross-device generalization. Self-supervised learning offers a promising direction, but current techniques fail to address the unique characteristics of gesture data—shorter durations, non-repetitive movements, and gesture similarity.

H Appendix: Discussion and Limitations

Beyond Earbuds and Blind User Gesture Recognition: While *GestureLens* demonstrates robust performance with earbuds and smartwatches, we envision expanding the system to emerging wearable form factors. Smart rings provide continuous finger and hand motion tracking with minimal user burden, while VR headsets and AR glasses enable head-mounted sensing. Beyond device diversity, we plan to evaluate *GestureLens* with populations with distinct motor characteristics, particularly individuals with Parkinson's disease whose tremor patterns and movement dynamics vary significantly across users. Such populations may benefit from gesture-based interfaces that accommodate motor variability. Additionally, our current evaluation focuses on indoor environments with minimal ambient noise. In the future, we will evaluate *GestureLens*'s robustness under real-world deployment scenarios, including noisy environments, moving vehicles, and outdoor settings.

Exploring Gestures across Modalities: Our framework currently focuses on IMU data, offering advantages in privacy, power consumption, and device availability. In the future, we will extend *GestureLens* to extract human motion representations from other modalities such as surface electromyography (sEMG) and camera-based vision systems. Rather than combining multiple modality streams, our approach would leverage a unified motion model that can process various input modalities independently, maintaining the same underlying representation of human motion regardless of the sensing technology used.

Transformer Architectures for Time-Series Signals: While *GestureLens* demonstrates strong performance for gesture recognition, transformers may not be optimal for continuous time-series signals despite our embedding of the motion signature (Zeng et al., 2023; Chen et al., 2023). In future work, we plan to explore specialized architectures like Conformers that combine convolutional networks with transformers to better model local patterns while maintaining global context. These architectural methods could further improve disambiguation between subtly different gestures with similar motion patterns.

Testing on Real-world Datasets: Our current system is evaluated mostly on controlled, noise-free datasets, and its performance in real-world noisy settings (e.g., outdoor environments or moving conditions) is not fully explored. Some of the datasets we use have a limited number of gesture classes and relatively small user pools, which may not capture the full variability seen in practice. In the future, we plan to evaluate *GestureLens* on larger and more diverse datasets, include more complex and fine-grained gestures, and test robustness in real-world conditions and across different wearable devices. Also, we acknowledge that our systems currently depend on manually defined descriptions and template design for classification. In the future, we will explore methods to make it generalizable.