ActivitySeeker: Towards Collaborative Personalized Human Activity Discovery and Recognition on Smartphones

Zhoutong Ye*
Department of Computer Science and Technology, BNRist
Tsinghua University
Beijing, China
yezt24@mails.tsinghua.edu.cn

Yanwen Huang*
Department of Computer Science and Technology
Tsinghua University
Beijing, China
huangyw21@mails.tsinghua.edu.cn

Chun Yu†
Department of Computer Science and Technology, BNRist, College of AI
Tsinghua University
Beijing, China
chunyu@tsinghua.edu.cn

Yuntao Wang
Department of Computer Science and Technology, BNRist
Tsinghua University
Beijing, China
yuntaowang@tsinghua.edu.cn

Yuqi Luo
Department of Computer Science and Technology
Tsinghua University
Beijing, China
luoyq21@mails.tsinghua.edu.cn

Yuanchun Shi†‡
Department of Computer Science and Technology, BNRist
Tsinghua University
Beijing, China
Qinghai University
Xining, China
shiyc@tsinghua.edu.cn

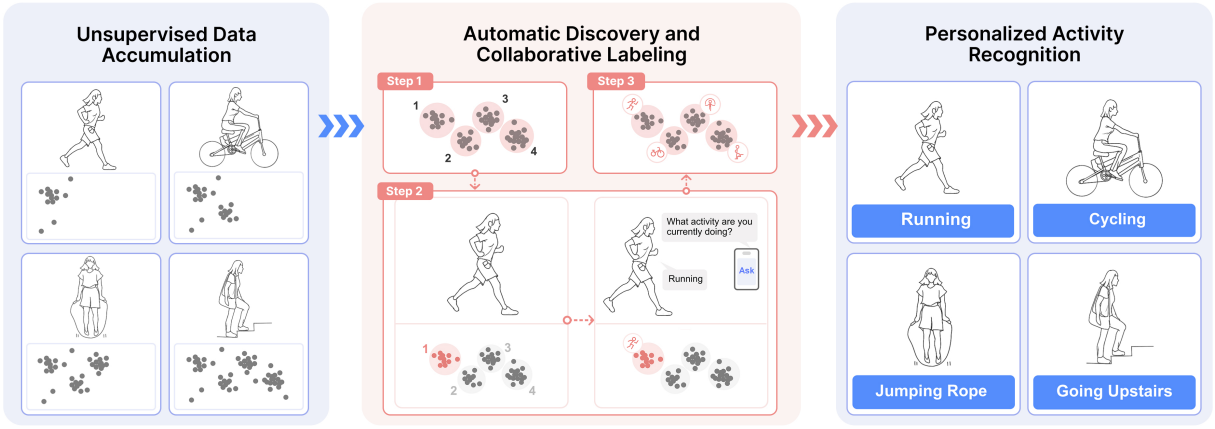


Figure 1: Overview of ActivitySeeker. The system passively accumulates smartphone IMU data during the user's daily activities. Through clustering, potential new activities are identified and then presented to the user for collaborative labeling, facilitating the construction of a labeled personalized dataset. With continuous usage, ActivitySeeker achieves high accuracy in personalized activity recognition. This user-centric approach enables the system to adapt and improve over time.

Abstract

Smartphones provide an attractive yet challenging platform for human activity recognition (HAR). They are ubiquitous, but also limit the input of HAR systems to a single IMU. These systems are also challenged by the inherent diversity of human activities and varying phone placement on the user's body. This results in traditional smartphone HAR systems having limited personalization potential or imposing a high user burden. We propose ActivitySeeker, a personalized smartphone HAR system that combines self-supervised activity discovery and low-burden user interaction to collaboratively label IMU data and adapt HAR models to individual users on-device through transfer learning. We evaluated ActivitySeeker through simulated online learning and in-the-wild user experiments, where it discovered 95.5% of personal activity types and achieved high recognition accuracy (93.3%) while maintaining a positive user experience. Leveraging the synergy between user and smartphone, ActivitySeeker opens up new possibilities for HAR-based applications like fitness, health and personalized recommendation.

*Both authors contributed equally to this research.
†Also with Key Laboratory of Pervasive Computing, Ministry of Education
‡Corresponding Author

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3791014

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Keywords

human activity recognition, automatic class discovery, personalization

ACM Reference Format:
Zhoutong Ye, Yanwen Huang, Chun Yu, Yuntao Wang, Yuqi Luo, and Yuanchun Shi. 2026. ActivitySeeker: Towards Collaborative Personalized Human Activity Discovery and Recognition on Smartphones. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3772318.3791014

1 Introduction

Human activity recognition (HAR) is an active field of research. It is essential for health monitoring, fitness, recommendation systems, smart assistants, and a wide range of other applications. Smartphones are an attractive platform for HAR. They are ubiquitous, possess essential sensors and computing power, and are frequently carried around by the user during various activities. Specifically, HAR systems based on smartphone inertial measurement units (IMUs) have shown great potential [6, 7, 17, 55], and could unlock various HAR-based applications with little extra cost for the user.

However, the smartphone also poses significant challenges as a platform for HAR systems. The single onboard IMU limits the amount of information available to the HAR algorithm. Moreover, compared to sensors with fixed placement (e.g. smartwatches, wearable sensors, etc.), variations in the placement of the phone on the user's body further complicate the design of HAR systems. Finally, the varying habits and needs of users demand the HAR system to adapt to each user and offer personalized activity recognition capabilities. Three factors are in play here: cross-user variance stems from the fact that different users perform the same activity (e.g. walking) differently. Within-user variance is rooted in factors like fatigue, clothing and the environment, which cause the same user to perform the same activity differently in separate *trials* (a *trial* refers to an entire episode of an activity, like *walking* from the office to the metro station). The third and final factor is the varying needs of users, who may want the HAR system to recognize different types of activities. For instance, an elderly user may prioritize the recognition of activities of daily living (ADLs) and fall detection, while a young, active user may want the system to individually recognize different activities in their workout session (e.g. push-up, pull-up, running, etc.). In essence, practical smartphone HAR systems need to adapt to different users and circumstances, providing personalized HAR capabilities while overcoming the limits posed by the lack of sophisticated sensors.

Current research on personalized HAR does not offer a practical path to adaptive smartphone-based HAR systems. Some prior works train personalized HAR models on manually labeled personal datasets [1, 10, 39, 60]. While these models achieve good performance, the data collection process itself is too burdensome and impractical for everyday use [45, 54]. Meanwhile, self-supervised and semi-supervised approaches reduce user burden at the cost of worse HAR performance [44, 62]. Moreover, a considerable portion of prior work is based on datasets collected in labs, failing to fully reflect the diversity of human activities in-the-wild. Finally, research into the recognition of user-specific activity types not seen in the training dataset is relatively lacking.

In this work, we introduce ActivitySeeker, a system designed to discover, label, and recognize personalized activities. It overcomes the challenges posed by limited sensing capabilities and highly variable environments through collaboration with the user. As a result, our system is capable of discovering and recognizing both common and rare, user-specific activities accurately, while causing minimal user burden. Specifically, ActivitySeeker adapts to a new user in three phases. In the discovery phase, ActivitySeeker continuously collects unlabeled smartphone IMU data in the background, which is then mapped by a pre-trained embedding model into a feature space and grouped into clusters in a self-supervised manner. In the labeling phase, when the user's current activity matches one of the unlabeled clusters, ActivitySeeker prompts the user to provide a label, which is then used to annotate the entire cluster. Over time, the collaborative labeling process yields a labeled personalized activity dataset. In the final phase, ActivitySeeker learns to recognize the activities in the personalized dataset through transfer learning, training a lightweight personalized model on-device. This eliminates performance degradation caused by cross-user variance. By repeating the whole process, ActivitySeeker gradually learns to recognize most of the user's activity types, and adapts to the within-user variance of each activity type. Our study results show that this approach can lead to activity recognition capabilities similar to HAR systems trained on manually labeled personalized data, while delivering user experience on par with the Apple Watch. In summary, the main contributions of this paper are:

- We propose ActivitySeeker, an end-to-end system that combines self-supervised learning, effective user interaction and transfer learning to discover and recognize human activities. The proposed system is designed to address cross-user and within-user variance, and be efficiently deployed on smartphones.
- We evaluated ActivitySeeker's performance on real-world data from 13 free-living participants. In the simulated online learning experiment, the system successfully discovered 95.5% of activity classes, correctly labeled 98.8% of unlabeled data segments in the collaborative labeling process, and achieved an average recognition accuracy of 93.3%, significantly outperforming baselines.
- We conducted two in-the-wild user studies to evaluate the user experience and activity recognition performance of ActivitySeeker. Our system significantly reduced physical and mental burden for users compared to manual labeling. When compared to the Apple Watch, users rated ActivitySeeker higher in terms of activity discovery and recognition capabilities and roughly the same in terms of user experience.

2 Related Work

2.1 Human Activity Recognition

Human activity recognition (HAR) refers to the detection and classification of human activities [2, 11, 54]. HAR has been extensively studied using various hardware, input modalities, and machine learning models. Wearable sensors, video, audio, RFID tags, and

environmental sensors and other modalities have been used in HAR tasks [50, 57, 68, 69, 71, 73, 77, 79]. While these approaches showed promise in experiments, they may be too intrusive for daily use, raise privacy concerns, have limited range, and require costly additional infrastructure [19, 29].

In contrast, smartphones provide an attractive platform for HAR. They are ubiquitous, rich in sensors, and possess considerable processing power which enables on-device model deployment [38]. As an integral part of users' daily lives, the smartphone also enables sensing tasks while causing minimal intrusion. As a result, smartphone-based HAR is well-suited for deployment in daily life and has been extensively investigated [6, 7, 41, 49].

A variety of models have been used in HAR, ranging from random forest and SVM to deep learning architectures like CNN, LSTM and transformers [2, 13, 16, 23, 30, 32, 33, 35, 43, 46, 51, 54, 81]. Further analysis demonstrates that features learned by deep neural networks (DNNs) are more generalizable than handcrafted statistical features and can be used in other activity recognition tasks [46, 53, 70, 77, 78, 80], which is further backed up by recent developments in autoencoders and contrastive learning models trained on readily available unlabeled data [4, 24, 31, 35, 79]. In ActivitySeeker, we employ a ResNet-inspired model structure [26, 34], which is capable of extracting features that can generalize to new users and custom activity types, while being lightweight enough for efficient on-device deployment.

HAR ranges from recognizing low level activities like walking and running to complex activities like cooking and playing football. This paper, in particular, deals with atomic activities, which are distinguishable by unique motion patterns and cannot be further decomposed into simpler activities [40, 47]. Examples include walking, running and doing push-ups. Existing smartphone IMU HAR datasets (e.g. MotionSense [41], HHAR [66], UCI-HAR [6], etc.) focus on a limited set of atomic activities, namely walking, cycling, running, ascending and descending stairs, standing, sitting, and lying down. In addition, the phone placement in these datasets was fixed, which fails to address the natural wearing diversity [14] of smartphone IMU data. In ActivitySeeker, we push beyond this limited set and tackle wearing diversity head-on by collecting a large (112.8 hours) in-the-wild dataset from free-living users and developing an effective HAR pipeline.

2.2 Personalization in HAR

Personalization is an important aspect of HAR, and has been extensively investigated [44, 55, 78]. Previous research has revealed significant cross-user variance, causing performance degradation when pre-trained models are used off-the-shelf by new users [5, 39, 62]. Various approaches to personalization have been studied. One solution involves fine-tuning pre-trained models with manually labeled data from end users [1, 10, 39, 60]. This approach achieves strong performance but requires a substantial amount of manual input from users, significantly elevating user burden. Moreover, obtaining user-labeled data in real-life scenarios is not always feasible [45, 54]. Other methods circumvent this challenge by leveraging unlabeled data [44, 62] but offer limited performance improvement and generally cannot recognize user-defined custom activities. Previous works introduced hybrid approaches where a small number of samples are labeled by the user in an interactive process, aiding the utilization of unlabeled data [8, 27, 77]. These hybrid approaches showed promise and inspired the design of ActivitySeeker.

Finally, to enable in-the-wild personalized HAR on smartphones, the problem of *wearing diversity* [14] must be addressed. Wearing diversity refers to the different placements and orientations of smartphones on users' bodies. For instance, a user may walk while holding their phone, as well as placing it in their coat pocket or trouser pocket. Research in this regard is relatively lacking, as most publicly available IMU-centric datasets are collected from sensors with fixed placement. This influenced our decision to collect our own dataset from free-living participants.

We address these limitations by proposing a system that accurately labels personalized IMU data through a collaborative process with low user burden, resulting in robust personalized HAR models that evolve with use.

2.3 Related Machine Learning Techniques

Our system aims to train personalized HAR models with minimal user input. Few-shot learning has shown promise by enabling the recognition of previously unseen, user-defined custom classes using a few labeled samples [52, 78]. An effective few-shot learning approach involves automatically annotating unlabeled samples using available labeled shots [72]. We aim to improve the model's performance as data accumulates continually, putting our work into the domain of incremental learning and continual learning [18, 44, 77]. Furthermore, users may perform activities not seen in the train set, necessitating novel class discovery (NCD) capabilities [22, 28, 61].

To minimize user burden, the discovered clusters undergo a mostly automated labeling process. Several methods have been proposed to facilitate automatic labeling in HAR and beyond, including label propagation [8, 20], manifold learning [62], using existing classifiers [15], and integration of other input modalities [52, 56]. The timing of user interactions is studied in prior work, which use active learning techniques to minimize user burden and maximize the knowledge gained from interactions [8, 27, 42, 45, 67]. Additionally, we employ transfer learning, which refers to the transfer of knowledge learned in one task to another [74, 85]. Transfer learning is widely used in Computer Vision (CV), Natural Language Processing (NLP), and HAR [42, 44, 60, 65]. Leveraging pre-trained models for feature extraction, transfer learning eliminates the need to train models from scratch, significantly reducing computation costs. The lightweight nature of transfer learning allows us to train personalized HAR models on-device.

2.4 Automatic Discovery and Recognition of Acoustic Events

ListenLearner [77] applies the same high-level concepts as ActivitySeeker does to acoustic activity discovery and recognition. It uses Ward's method to cluster audio events based on feature representations generated by a VGG-ish embedding model. For each cluster, a one-class SVM [64] classifier is trained for the corresponding activity.

Compared to the audio modality, smartphone IMU poses several unique challenges. The first and most important is **the large cross-**

and within-user variance. Audio events are largely consistent over time - for example, microwaves sound roughly the same at different places and times. This is further aided by the fact that smart speakers like ListenLearner are rarely moved around and deal with largely unchanging environments. In contrast, as discussed in Section 2.2, motion patterns captured by smartphone IMUs are far more variable. For the task of personalized HAR, within-user variance due to factors like different clothing, wearing diversity [14], and fatigue levels mean that ActivitySeeker has to manage far more clusters for each type of activity, making the task more difficult than self-supervised acoustic event recognition. The second challenge is **managing noise**. ListenLearner uses a simple threshold of 1.5 standard deviations above ambient sound levels to filter out low-information segments. For IMU data, this approach would lead to many meaningful activities with smaller or slower movements (e.g. doing push-ups with the phone in one's pocket) being discarded. Therefore, ActivitySeeker needs a "smarter" way of noise filtering. Finally, **on-device deployment** is much more challenging for smartphones. ListenLearner is a specialized device for acoustic event recognition plugged into a power outlet, while ActivitySeeker needs to minimize the computational cost and battery drain on a smartphone that also hosts other applications.

The first challenge means that ActivitySeeker cannot use the "one cluster per activity" approach of ListenLearner. Instead, each label may correspond to multiple clusters in ActivitySeeker. Therefore, clustering and training personalized classifiers are done separately in our pipeline. By decoupling the learning of local structures (i.e. clusters) and global structures (i.e. activities), ActivitySeeker is more versatile and able to effectively handle the discovery and recognition of ever-changing human activities. To further tailor ActivitySeeker to IMU input, we designed a hybrid DNN encoder that incorporates both temporal and frequency domain inputs, enabling the extraction of finer-grained features that accurately capture the local structures in the space of human activities. To deal with the second challenge (noise filtering), we use a decision tree model that considers both periodicity and magnitude to filter out noise while retaining low-magnitude segments with high periodicity. Finally, we optimized the lightweight pretrained encoder, clustering process, and personalized SVM classifier to run efficiently on a smartphone with minimal latency and battery usage. This is different from ListenLearner, which relied on a server for training personalized models.

2.5 Comparison with Commercially Available Products

ActivitySeeker stands out among commercially available solutions by combining the ability to discover and recognize user-defined custom activities, the ability to adapt to a user over time, and low interaction burden. The Apple Watch, for instance, can only automatically recognize 9 predetermined types of workouts. For custom workouts, users have to manually log the beginning and end of each session. Fitness applications like Strava and Keep require the user to manually start a workout recording, though Strava has incorporated the auto-pause feature to reduce user burden. In comparison, ActivitySeeker is more versatile and less burdensome for users.

3 Method

3.1 Learning Feature Representation of IMU Data

The basis of any HAR system is the ability to extract features from sensor data. We trained a ResNet based model on a diverse smartphone IMU dataset, using the feature representation learned by the model as the basis of ActivitySeeker.

3.1.1 Pre-training Dataset. The pre-training dataset combines IMU data from the public MotionSense dataset [41] with data collected from 4 free-living subjects over two weeks in our own experiments. MotionSense, with data collected from 24 users in a short, controlled study, primarily captures cross-user variance. Our own data, collected over a longer time frame (around 2 weeks) from free-living users, focuses on within-user variance and complements MotionSense. The performance impact of introducing our own data is shown in Appendix C. The content of the combined dataset is shown in Table 1.

Activities	Specified Positions
Static*	Trouser Pocket (N=3325), Holding with Operations (N=250)
	Placed on Surfaces (N=552)
Walking*	Trouser Pocket (Fast, N=223), Trouser Pocket (Slow, N=2878)
	Holding (N=419), Holding with Operations (N=212)
	Coat Pocket (N=183)
Going Upstairs*	Trouser Pocket (N=1029)
Going Downstairs*	Trouser Pocket (N=744)
Jumping Rope	Trouser Pocket (N=48)
Jumping Jack	Trouser Pocket (N=32)
Running*	Trouser Pocket (N=861), Holding (N=281), Coat Pocket (N=18)
Cycling	Trouser Pocket (N=1238)
Riding E-bike	Trouser Pocket (N=129)
Squatting	Trouser Pocket (N=85)
Push-up	Trouser Pocket (N=43)
Leg Shaking	Trouser Pocket (N=399)

Table 1: The content of the Pre-training Dataset, including data from both MotionSense and our experiment. **Activity classes present in both MotionSense and our experiment.*

In our data collection experiment, four subjects from a university campus participated (1 female and 3 males, aged 21-23, average age 22), using their personal Android smartphones equipped with a data collection app. This app enabled them to choose an activity label, start, and stop activity recordings. We set the IMU sampling rate at 100Hz, high enough for classification performance across various HAR datasets [36]. The subjects were asked to record activities in their daily routines and upload the collected data at the end of each day. In total, 10.8 hours of IMU data was collected over two weeks, encompassing 20 labels (we put the same activity with different phone placement under separate labels). Additionally, the MotionSense dataset added 7.8 hours of smartphone IMU data to our pre-training dataset, bringing with it a significant amount of cross-user variance.

For data collected in our own experiments, we manually screened the data for abnormalities (e.g. users forgetting to turn off the app after finishing an activity, IMU turned off in power-saving mode, etc.). The data from the MotionSense dataset is used off-the-shelf and resampled to 100 Hz. The combined dataset is then divided into

5-second-long windows with no overlap. To further improve the pre-trained model's robustness, we applied time warping and scaling [75] to the dataset. Time warping introduced variations in the speed at which the activity is performed while scaling introduced variations in the intensity of the activities. The data augmentation process increased the amount of training data to 4 times the original amount. See Appendix B for details on data augmentation.

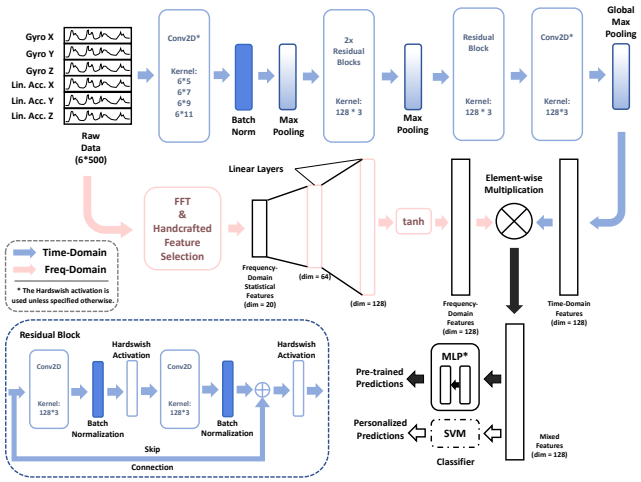


Figure 2: Model architecture of ActivitySeeker's feature extraction model. The mixed temporal and frequency feature vector is fed into either a pre-trained MLP or a personalized SVM prediction head for activity predictions.

3.1.2 Model Architecture. We observed that both the time and frequency domains contain valuable features (See Appendix I), in line with the results shown in [21, 70]. As a result, our model processes both time domain data and frequency domain features. The time domain data is a 5-second window of triaxial linear accelerometer and triaxial gyroscope data, sampled at 100Hz. The 5-second time window was chosen because it sufficiently covered the period or duration of most atomic activities. The 100Hz sampling rate was chosen to capture detailed motion patterns [36]. The frequency domain data consists of 20 handcrafted features including the periodicity score (the ratio between the highest peak in the spectrum and the 75th percentile, a simple periodicity metric similar to the ones used in [48, 82], 1 dimension), the overall mean energy of the 6 spectra (1 dimension), a one-hot encoding of the input channel with the highest periodicity score (6 dimensions), the log of the energy of each axis' spectrum (6 dimensions), and the dominant frequency of each axis (6 dimensions), forming a 20-dimensional feature vector. We chose these features empirically after observing the visualized FFT spectra of different activities (examples are in Appendix I).

As shown in Figure 2, we use ResNet [26] as backbone of our model. The ResNet extracts features from the time domain, while a multilayer perceptron (MLP) processes frequency domain features. Previous works have shown that combining time and frequency domain features enhances overall model performance [21, 70]. In our case, fusing the time and frequency domain feature vectors through element-wise multiplication yielded the best result. This is shown in Table 2.

3.1.3 Performance. We trained and evaluated our model and the baselines on the pre-training dataset described in Section 3.1.1. We split the dataset into 70% for training, 15% for validation, and 15% for testing. All splits were done in a within-user manner, where the data from a user may appear in all three splits, although there was no overlap between the splits. We did not use a cross-user setting here because, to train a feature extractor, it is more productive to use the pre-training dataset to the full extent and train on data from all users. The best-performing model on the validation set was saved and evaluated on the test set. The results over 5 different random number seeds, presented in Table 2, show that the ResNet backbone performed significantly better than other backbones like CNN and LSTM. The fusion of time and frequency domain features, as well as data augmentation, improved model performance. These results validate our model architecture and provide a solid foundation for ActivitySeeker.

Model	Accuracy	F1-Score	p-value
Time-Only CNN	$0.937_{\pm 0.007}$	$0.919_{\pm 0.017}$	0.0016 (**)
Time-Only LSTM	$0.929_{\pm 0.011}$	$0.869_{\pm 0.022}$	0.0005 (***)
Time-Only ResNet	$0.975_{\pm 0.001}$	$0.968_{\pm 0.003}$	0.0308 (*)
Time-Freq Concat.	$0.980_{\pm 0.002}$	$0.971_{\pm 0.007}$	0.0774
Time-Freq Mult. w/o Data Aug.	$0.974_{\pm 0.002}$	$0.959_{\pm 0.007}$	0.0050 (**)
Time-Freq Mult. (ActivitySeeker)	$\mathbf{0.981}_{\pm 0.002}$	$\mathbf{0.976}_{\pm 0.002}$	N/A

Table 2: Activity classification performance of different model structures. We performed a paired t-test on the macro-F1 results of each alternative model and the model used in ActivitySeeker. * denotes $p < 0.05$, ** denotes $p < 0.01$, * denotes $p < 0.001$.**

3.2 Enabling Personalized HAR from Scratch

ActivitySeeker works in three phases. In the first phase, it collects and segments unlabeled IMU data in the background. It filters out static windows while mapping active windows into the feature space using the pre-trained embedding model. In the second phase, the feature vectors are clustered, and the unlabeled clusters are then annotated through collaborative labeling. The third and final phase involves training a lightweight, personalized model through transfer learning, enabling the precise recognition of the user's activities. The pipeline is shown in Figure 3.

3.2.1 Data Segmentation and Noise Removal. In ActivitySeeker, IMU data is sampled at 100Hz and segmented into 5-second windows with 50% overlap. The overlap doubles the samples available, mitigating the scarcity of personalized data. Recognizing that smartphone IMU data is inherently noisy, we distinguish between *active* and *static* windows. *Active* windows exhibit high energy or periodicity, typical of dynamic activities like walking or cycling, whereas *static* windows lack dominant frequency and have low energy. We focus solely on active windows, filtering out static ones. We used a decision tree classifier to filter out static windows based on statistical features like periodicity and energy from linear acceleration and gyroscope data (see Appendix A for details). This classifier successfully differentiates active windows from static ones with

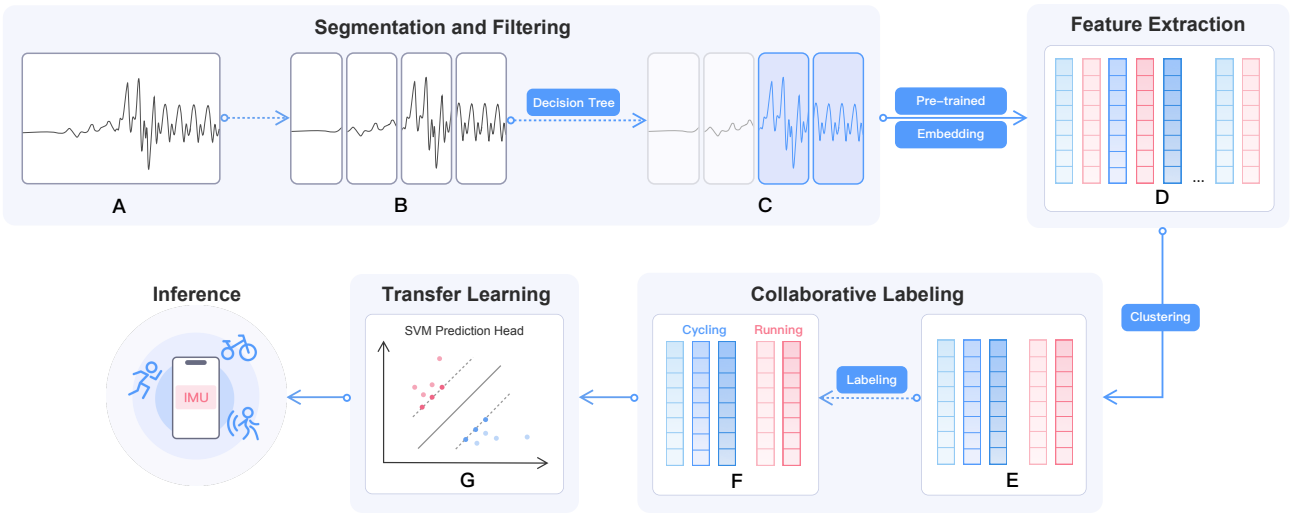


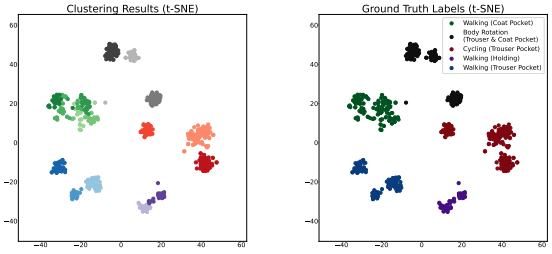
Figure 3: An overview of the ActivitySeeker pipeline. The raw data is preprocessed to filter out static windows and then mapped to a feature space using a pre-trained embedding model. Clustering is employed to discover potential activities, and collaborative labeling is utilized to create a personalized dataset. The user's own SVM prediction head is trained through transfer learning, facilitating the recognition of personalized activities.

over 90% accuracy, reducing the volume of data for subsequent processing stages.

3.2.2 Feature Extraction and Clustering. Following the filtering stage, feature vectors are extracted from active windows using the pre-trained embedding from Section 3.1. We then use agglomerative hierarchical clustering with complete linkage and cosine similarity distance metric to reveal activity patterns. The clustering process halts when the average cluster size reaches a predefined value ($\bar{S}_{cluster} = 13.5$). We demonstrate the effectiveness of this clustering approach in Figure 4, where each activity forms distinct clusters.

3.2.3 Collaborative Labeling. We propose a collaborative labeling approach to reduce user burden while accurately annotating a large amount of data. Specifically, when the user's current activity matches one of the unlabeled clusters, the system prompts the user to label the activity *in situ*. This label is then used to annotate the entire cluster. A k-NN classifier is used to match the current activity to a cluster. Since k-NN alone is susceptible to outliers, we model the cluster using a one-class SVM [63] to double check whether the current activity indeed belongs to the cluster. The system will ask the user for a label if the SVM yields a positive result. Section 5.1 provides details on how we implemented the interaction process of collaborative labeling.

Collaborative labeling combines the system knowledge gained through clustering and the user's *a priori* knowledge about their own activity, enabling effective human-machine collaboration in annotating IMU data. By annotating an entire cluster of recorded past events with a single interaction, the amount of manual input needed is significantly reduced.



(a) Clustering Result **(b) Ground-truth Labels**

Figure 4: 2-D t-SNE visualization of clustering results. Clusters of the same ground truth label use different shades of the same color. Notably, no overlap is observed between clusters belonging to different labels, ensuring clear differentiation and accurate data representation. *Body Rotation* was not included in the pre-training dataset (described in Table 1), but the system successfully discovers this previously unseen activity. It can be seen that each activity type is subdivided into several clusters, reflecting the within-user variance of human activities.

3.2.4 Training Personalized Models. Leveraging the feature vectors extracted by the pre-trained model, we train a personalized multi-class SVM classifier with radial basis function (RBF) kernel. The training process is fast and energy efficient, making it suitable for mobile deployment. The model undergoes incremental updates as the volume of labeled data increases, improving both its accuracy

and widening the range of activities it recognizes. We set a threshold of 500 newly annotated samples (N_{update}) for each update cycle, a figure determined empirically to balance between minimizing computational load and ensuring timely model updates. Upon reaching this threshold, the SVM is retrained with the expanded dataset.

4 Quantitative Evaluation

In this section, we will discuss the quantitative evaluation of the system's performance through simulated online learning. We collected a substantial real-world dataset from participants engaged in uncontrolled, free-living scenarios to facilitate the quantitative evaluation.

4.1 Simulated Online Learning

ActivitySeeker, designed to label past activities automatically, is difficult to evaluate quantitatively in the wild due to the absence of ground truth activity labels. Furthermore, alternative methods of obtaining labels during the experiment, such as participant recollection or audio and video recordings, face reliability issues or pose significant privacy and logistical challenges. Therefore, deploying the pipeline on smartphones and conducting an in-the-wild user experiment cannot, on its own, offer a quantitative evaluation of ActivitySeeker's performance.

To address this, we adopted a simulated online learning approach, where users labeled their activities during their daily lives, creating a dataset that was then fed into the system in chronological order, simulating the real-world data flow. The ground truth label given by the user is hidden, and the collected data is fed into the system as unlabeled samples. When the model discover that the current activity matches a cluster of past activities in the collaborative labeling phase, the ground truth label of the current activity is used as the answer to the question, "What activity are you currently doing?". This label is then used to annotate the entire cluster. This simulated online learning experiment, similar to those done in recent research [3, 12, 25], effectively recreates real-world online learning scenarios.

At the end of the simulation, much of the data is labeled automatically by ActivitySeeker through simulated collaborative labeling. Since ActivitySeeker is not 100% accurate when annotating unlabeled data, the collaborative labeling process will result in a small amount of mislabeled samples. By comparing the label given by ActivitySeeker with the ground truth label, we can assess the accuracy of the collaborative labeling process quantitatively.

4.2 Real-World Dataset

4.2.1 Rationale. Existing open-source IMU-centric datasets have several limitations, which make them unsuitable for the quantitative evaluation of ActivitySeeker through simulated online learning. Specifically, these public datasets have limited variation in phone positioning, cover a narrow range of activity categories, or tend to capture each user's activities briefly, typically for less than 5 minutes per activity per user [6, 7, 9, 37, 57, 59]. Moreover, these datasets are generally collected in controlled environments, which may not accurately reflect users' daily routine. For instance, many datasets equally represent activities like walking and climbing stairs, despite walking being far more common in everyday life. Therefore, we collected real-world IMU data from free-living participants over an extended period (8-20 days) to quantitatively test ActivitySeeker on a dataset that closely resembles an actual deployment scenario.

In total, we collected 112.8 hours of labeled non-static IMU data from 13 free-living users (4 females and 9 males, ages 19-26, average age 22) in a university campus, averaging 8.7 hours per user. This is an order of magnitude greater than the 0.3 hours per user collected in MotionSense [41] and 0.6 hours per user in PAMAP2 [57]. The rationale behind the greatly increased data per-person is twofold: (1) ActivitySeeker passively collects unlabeled IMU data in the background while users carry on with their daily lives normally, allowing for a much larger amount of within-user data, and (2) our focus on personalized HAR necessitates more data per person to effectively capture the within-user variance. The detailed data collection procedure is available in Appendix D. To prevent data leakage and ensure fair evaluation, the 4 users who contributed to the pre-training dataset were not included in this dataset.

4.2.2 Dataset Composition. In total, the real-world dataset we collected contained 15 types of activities, including those not covered by the pre-training dataset. The names of the activities and the number of users performing each activity are shown in Table 3. More detailed information about the composition of this dataset can be found in Appendix H. In terms of activities covered, this dataset is much more diverse than existing smartphone IMU datasets such as MotionSense [41], UCI-HAR [6] and HHAR [66], which only covers walking, ascending and descending stairs, cycling, jogging, and static behaviors like sitting. With more than 10 hours of in-the-wild IMU data per user, our dataset also does a better job covering the within-user variance.

Activity	#Users	Activity	#Users
Walking (Holding)	13	Running (Holding)	10
Walking (Coat Pocket)	8	Running (Coat Pocket)	2
Walking (Trouser Pocket)	11	Running (Trouser Pocket)	5
Going Upstairs (Trouser Pocket)	11	Cycling (Trouser Pocket)	13
Going Downstairs (Trouser Pocket)	11	Squatting (Trouser Pocket)	4
Jumping Rope (Trouser Pocket)	5	Push-up (Trouser Pocket)	4
Jumping Jack (Holding)	7	Pull-up (Trouser Pocket)	1
Body Rotation (Trouser & Coat Pocket)	9		

Table 3: Activities encountered in the user experiment.

4.3 Experiment Setup

ActivitySeeker was evaluated through the simulated online learning method described in Section 4.1. Specifically, 80% of the data from each user's activity recording was allocated for system training. This data is sent into the system in chronological order according to timestamps during simulated online learning. The remaining 20% of data were used for testing. We guaranteed that each trial was used for either training or testing, but not both. This prevented data leakage, and maximized the within-user variance between the train set and test set. In this experiment, we report macro F1 (mF1) in addition to accuracy (Acc). Since human activities have a natural class imbalance, we follow the precedent set by recent work in HAR [81, 83, 84] and use mF1 as the primary metric.

We compared ActivitySeeker to a baseline model with no personalization, a baseline model using conventional continual learning

	Baseline 1		Baseline 2		ActivitySeeker			Optimal	
User	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**C**	**Accuracy**	**F1-Score**
1	**0.857**	0.686	0.531	0.184	0.831	**0.784**	12.0/12	0.868	0.845
2	0.965	0.723	0.965	0.964	**0.983**	**0.985**	6.0/6	0.997	0.998
3	0.899	0.785	0.966	0.897	**0.981**	**0.978**	11.0/12	0.993	0.990
4	0.828	0.483	0.644	0.542	**0.854**	**0.818**	6.0/7	0.883	0.928
5	**0.953**	0.461	0.885	**0.936**	0.943	**0.936**	5.0/5	0.971	0.968
6	0.802	0.386	0.733	0.490	**0.905**	**0.732**	6.3/7	0.933	0.754
7	0.872	0.773	0.741	0.232	**0.890**	**0.786**	14.0/15	0.958	0.930
8	0.919	0.762	0.923	0.711	**0.926**	**0.895**	8.3/10	0.968	0.910
9	0.841	0.525	0.928	0.600	**0.977**	**0.937**	10.0/10	0.985	0.974
10	0.949	0.630	0.867	0.870	**0.980**	**0.976**	7.0/7	0.984	0.982
11	0.780	0.534	0.900	0.797	**0.942**	**0.903**	8.0/8	0.972	0.959
12	0.880	0.746	**0.972**	**0.942**	0.938	0.939	7.3/8	0.976	0.970
13	0.807	0.646	0.481	0.378	**0.973**	**0.969**	7.0/7	0.983	0.972
Avg	$0.873_{\pm 0.059}$	$0.626_{\pm 0.129}$	$0.810_{\pm 0.162}$	$0.657_{\pm 0.264}$	$\mathbf{0.933}_{\pm 0.048}$	$\mathbf{0.895}_{\pm 0.083}$	8.3 / 8.7	$0.959_{\pm 0.039}$	$0.937_{\pm 0.066}$

Table 4: Accuracy and macro F1-score for ActivitySeeker and the baseline methods. We also report the ratio of discovered classes to total classes (C) for ActivitySeeker. Since we ran the evaluation three times using different random seeds and averaged the results, the number of discovered classes may not be a whole number. C is the same for ActivitySeeker and baseline 2, but not applicable to baseline 1 and the optimal scenario, where all data is manually labeled.

algorithms (Replay+EWC)[18] for personalization, and an optimal scenario, all using the same test set:

- **ActivitySeeker: Personalized Data + Collaborative Labeling + Transfer Learning.** ActivitySeeker retains the pre-trained embedding and only retrains the SVM classifier on the personalized data, as shown in Figure 3. The training data for the SVM classifier is obtained through collaborative labeling.
- **Baseline 1: No Personalization.** Baseline 1 represents using pre-trained models off-the-shelf for new users. In our experiment, this is facilitated through the leave-one-user-out method. For each user, the whole model, as shown in Figure 2, is trained on all other users' manually labeled data.
- **Baseline 2: Personalized Data + Collaborative Labeling + Continual Learning.** Baseline 2 represents using existing continual learning algorithms to build personalized HAR models. It is based on two widely used continual learning techniques, replay and elastic weight consolidation [18], which are applied to the same pre-trained ResNet-based model as ActivitySeeker.
- **Optimal: Personalized Data + Manual Labeling + Transfer Learning.** We explore a hypothetical scenario with a 100% utilization rate and labeling accuracy of personalized data. This scenario is equivalent to having an ***oracle*** for automatic class discovery. The Optimal scenario represents the best possible model performance that can be derived from the personalized dataset of each user, and can be used to gauge the performance degradation caused by the collaborative labeling process and the trade-off between labeling accuracy and user burden.

4.4 Results

ActivitySeeker significantly outperformed both Baseline 1 (paired t-test, p=0.00001) and Baseline 2 (paired t-test, p=0.002), achieving better mF1. Since mF1 accounts for class imbalance, this result also highlights ActivitySeeker's ability to accurately recognize rarer activity types. In addition, ActivitySeeker performed reasonably well for all users, whereas the baselines performed extremely poorly for some users (e.g. Users 4, 5 and 6 for baseline 1 and Users 1, 7 and 13 for baseline 2) in terms of mF1. Furthermore, in this experiment, ActivitySeeker labeled 67.4% of samples with 98.8% accuracy on average, demonstrating the effectiveness of collaborative labeling. This is further supported by comparing ActivitySeeker with the hypothetical optimal scenario, which shows that collaborative labeling introduces minimal performance degradation compared to full manual labeling while significantly improving user experience (see Section 5).

We visualized confusion matrices from this experiment to better understand how ActivitySeeker worked for different types of users and activities. Figure 5 shows the results for a typical low-data user (User 8), a typical high-data user (User 3), and the combined results of all 13 users. For User 8, we can see that the limited personalized data for push-ups resulted in ActivitySeeker failing to discover the activity at all. In contrast, ActivitySeeker succeeded in discovering "push-ups" as a distinct activity for User 3, whose personal dataset is much larger. It is also evident that ActivitySeeker's performance improved as more data was accumulated - although User 3's data contained more types of activities, the overall performance is actually better than that of User 8. See Appendix H for the detailed composition of the users' personal dataset.

Looking at the combined results for all 13 users, we observe that wearing diversity (discussed in Section 2.2) remained challenging for ActivitySeeker: many of the mistakes were among the three labels where the user was running with different phone placements. Moreover, rarer activities like squatting and pull-ups suffered from the lack of data. High-intensity activities (running and jumping rope) also confused ActivitySeeker in some cases. In contrast, activities with ample data such as walking, cycling, going upstairs and going downstairs had much better results. This demonstrates that ActivitySeeker is pushing beyond the capabilities required in conventional smartphone IMU HAR tasks, as recognizing these

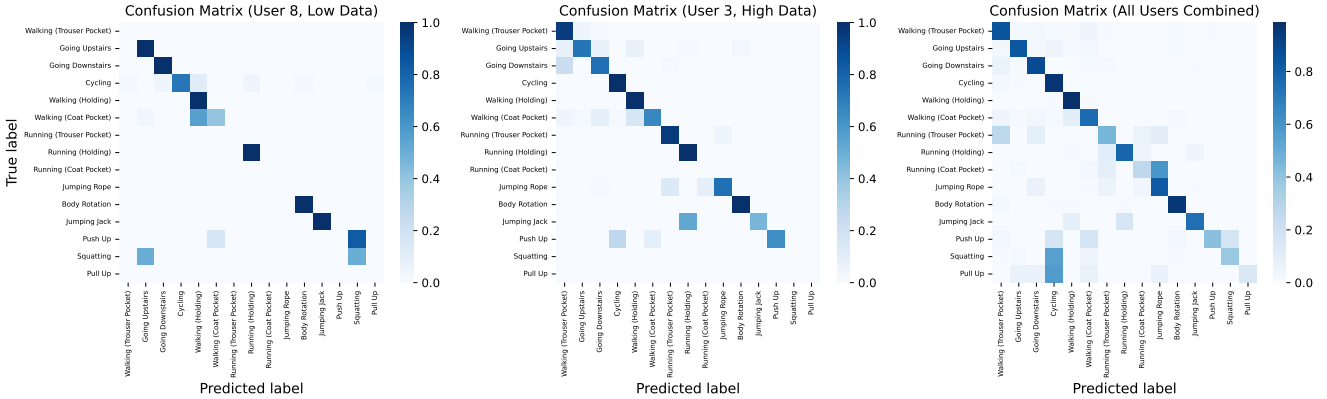


Figure 5: Confusion matrices from the simulated online learning experiment. We visualized the results for a user with limited personalized data (User 8), a user with more personalized data (User 3), and the combined results of all users.

activities is the core challenge of existing smartphone IMU datasets [6, 41, 58, 66].

4.5 Ablation Study

We conducted an ablation study to validate the design of ActivitySeeker. Since all three components are essential to the functioning of the pipeline, it was impossible to remove each component in the ablation study. Instead, we compare the performance of ActivitySeeker to that of pipelines with alternative components. In this section, we focus on clustering and transfer learning. For clustering, we tested different types of clustering algorithms (DBSCAN, k-Means, and Ward's). For transfer learning, we compared the performance of different lightweight classifiers (MLP, random forest, k-NN and linear SVM), as well as training a ResNet model from scratch on personalized data. The results are presented in Table 5. The ablation study results for feature extraction model can be found in Section 3.1.3 and Appendix C.

Component	Type	Accuracy	mF1
Clustering Algorithm	DBSCAN*	$0.802_{\pm 0.108}$	$0.730_{\pm 0.131}$
	K-Means*	$0.882_{\pm 0.111}$	$0.827_{\pm 0.165}$
	Ward's	$0.890_{\pm 0.092}$	$0.848_{\pm 0.124}$
	Agglomerative (ActivitySeeker)	$0.933_{\pm 0.048}$	$0.895_{\pm 0.083}$
Transfer Learning	Full Model (Retrain from Scratch)	$0.902_{\pm 0.083}$	$0.875_{\pm 0.087}$
	MLP*	$0.825_{\pm 0.103}$	$0.772_{\pm 0.123}$
	Random Forest	$0.908_{\pm 0.065}$	$0.881_{\pm 0.080}$
	k-NN	$0.920_{\pm 0.076}$	$0.890_{\pm 0.094}$
	Linear SVM	$0.920_{\pm 0.058}$	$0.890_{\pm 0.080}$
	RBF SVM (ActivitySeeker)	$0.933_{\pm 0.048}$	$0.895_{\pm 0.083}$

Table 5: Results of the ablation study, where we switch out pipeline components for alternatives. See Tables 2 and 8 for ablation results on the pre-trained feature extraction model. An asterisk (*) indicates a statistically significant (paired t-test, p<0.05) difference between the performance of ActivitySeeker and the alternative component in terms of mF1.

For clustering, agglomerative clustering held a clear advantage over other clustering algorithms. Using a suitable clustering algorithm contributed the most to ActivitySeeker's performance. In contrast, transfer learning worked well with every type of classifier except MLP, although SVM with RBF kernel achieved the best results. Leveraging the pre-trained feature extractor, lightweight classifiers were even able to outperform training the full model from scratch, demonstrating the effectiveness of transfer learning in personalized HAR. Based on the results, we recommend testing different types of classifiers and choosing the one that works the best when adapting ActivitySeeker to downstream applications. Overall, the results of the ablation study validate ActivitySeeker's design, and provide insight into adapting ActivitySeeker to downstream applications.

In addition to the algorithms and models used, we also explored the effect of the hyperparameter $\bar{S}_{cluster}$, which regulates the granularity of agglomerative clustering. Larger $\bar{S}_{cluster}$ results in larger, coarser-grained clusters, which lead to more data being labeled but also potentially increase the error rate of collaborative labeling. Smaller $\bar{S}_{cluster}$ results in smaller, finer-grained clusters, which lead to less data being labeled but increase the accuracy of collaborative labeling. We report the results in Table 6. On our dataset, $\bar{S}_{cluster} = 13.5$ led to the most balanced results. For downstream applications, we recommend tuning $\bar{S}_{cluster}$ to achieve the desired trade-off between the ability to discover new activities and the ability to recognize existing activities.

$\bar{S}_{cluster}$	Data Utilization	Collab. Labeling Acc.	C	Accuracy	mF1
10.0	0.552	0.993	7.7/8.7	$0.902_{\pm 0.064}$	$0.881_{\pm 0.069}$
13.5	0.674	0.988	8.3/8.7	$0.933_{\pm 0.048}$	$0.895_{\pm 0.083}$
17.0	0.747	0.978	8.5/8.7	$0.924_{\pm 0.076}$	$0.889_{\pm 0.110}$

Table 6: How $\bar{S}_{cluster}$ impacts the performance of ActivitySeeker. Data utilization refers to the proportion of input 5-second segments being assigned a label through collaborative labeling. Collaborative labeling accuracy measures how many machine-annotated labels that match the ground truth label. C, accuracy, and mF1 are the same as Table 4.

5 In-the-wild User Studies

We conducted two in-the-wild user studies to evaluate the usability of ActivitySeeker in real-world scenarios. In the first study, we aimed to compare the user experience of traditional manual labeling

and collaborative labeling. In the second study, we aimed to compare the user experience and HAR capabilities of ActivitySeeker and the Apple Watch. These experiments also offered a valuable opportunity to examine ActivitySeeker's performance under the resource constraints of a smartphone.

5.1 Implementing Collaborative Labeling In the Wild

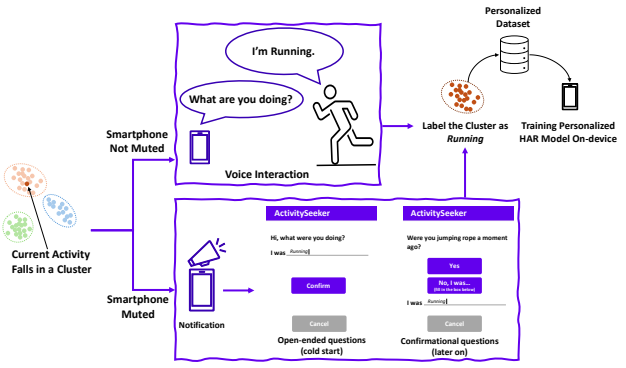


Figure 6: The interaction process of ActivitySeeker. We included two types of interaction - voice interaction when the phone is not muted, and a notification + GUI approach when the phone is muted.

A well-designed collaborative labeling process is crucial to providing a smooth, non-intrusive user experience. The ListenLearner study [77] showed that users prefer confirmation-style questions over open-ended ones. Therefore, we designed a three-stage strategy to lower user burden. First, when a user's current activity matches an unlabeled cluster, the system asks, "What activity are you currently doing?". After a while, if the current activity matches a labeled cluster, the system seeks confirmation with a question like "Are you running?". This is similar to how the Apple Watch alerts the user to record a workout, and lowers the frustration caused by prompting the user. Finally, after a reliable classification model is trained, the system confidently identifies known activities and informs the user with a message like "You are running." In such cases, the user does not need to provide an input unless the activity label is wrong. The interaction process of ActivitySeeker is shown in Figure 6. The UI of the manual labeling baseline and the Apple Watch can be found in Appendix E.

5.2 Collaborative Labeling vs Manual Labeling

5.2.1 Experiment Design. This study aimed to compare the user experience of collaborative labeling to that of manual labeling. We invited eight participants (4 females and 4 males, aged 20-26) for this study. All participants received the equivalent of 15 US dollars per hour for their time. We repurposed the manual labeling application developed to collect the pre-training and real-world datasets, and used it as the manual labeling baseline in this study. During the field study, ActivitySeeker and the baseline were used for 30 minutes each, running continuously throughout the entire period. A challenge in this experiment was the subconscious nature of many activities, which led to the user forgetting to manually label activities when testing the baseline application. Therefore, an observer accompanied the user during this study to objectively observe and remind the user to label activities. Despite the presence of an observer, the participants were free to choose the setting of the experiment and were encouraged to recreate part of their daily routine. The participants also selected and scheduled their activities as they wished, as long as these activities were atomic. Each participant was asked to choose at least two settings and perform at least five different combinations of activity and phone placement (see Appendix F for details). This enabled us to cover diverse real-world settings like offices, roads, libraries, dormitories, playgrounds and gyms.

When testing ActivitySeeker, the user carried their smartphone as usual, following their daily routine while the system was running continuously in the background. Throughout the experiment, the user could freely intermix different activities and transition from one scenario to another. The experimenter observed from the sidelines without interfering, recording the types of activities performed by the user. Participants were also free to pause for breaks if they became fatigued. By letting the system run throughout the 30-minute experiment, we could test the system's robustness to transitions between activities and noises caused by rest periods. When testing the baseline application, the user was instructed to perform activities freely but needed to manually start and stop data collection for each activity - in manual labeling, the beginning and end of each recording must be precise to maintain the data quality. Therefore, if the user forgot to start or stop recording when they transitioned to a new activity, the observer would remind them.

After completing the experiment, the users were asked to rate the physical burden, mental burden, time pressure, frustration, usability, and preference on a scale of 1 to 7, with 7 indicating the most positive experience. We also asked the users to rate the meaningfulness of activities discovered by ActivitySeeker. We wrapped up the experiment by conducting an open-ended interview to gain further insight on the user experience of ActivitySeeker.

5.2.2 Results. The results from the questionnaire are shown in Figure 7. We conducted statistical analysis using one-way ANOVA and reported the results in the figure. ActivitySeeker imposed significantly less physical burden, mental burden and time pressure, highlighting the benefits of collaborative labeling. The users rated the usability of ActivitySeeker highly, and showed a strong preference for ActivitySeeker over the baseline app. Finally, the activities discovered and recognized by ActivitySeeker were meaningful to the users, who gave an average meaningfulness rating of 6.5 out of 7.

The open-ended feedback session provided further context to the results from the questionnaire. As we had expected, manual logging is too burdensome for users, with one participant sharing that "I have to manually open the application, enter or select an activity, start, and stop it with the baseline. But with ActivitySeeker, I only respond to questions when necessary." On the subject of mental burden, most participants are satisfied with ActivitySeeker's performance. However, to our surprise, User 4 expressed slight anxiety when using ActivitySeeker, saying, "I keep thinking about when I will be asked next and want to respond immediately." This shows

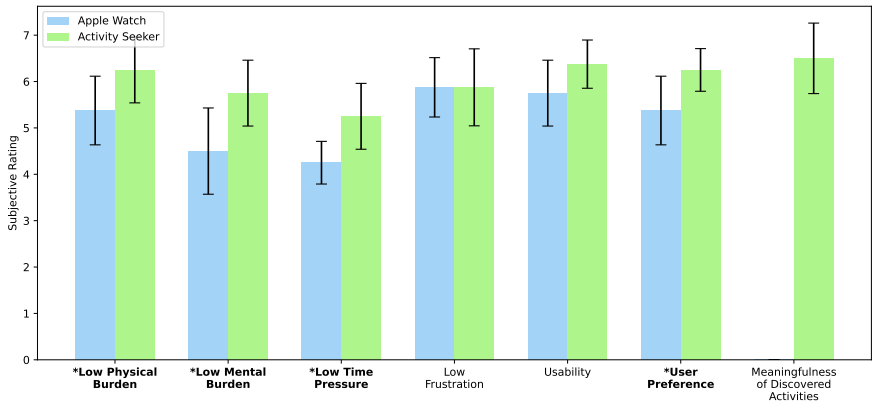


Figure 7: Subjective Ratings Between Baseline and ActivitySeeker. Higher ratings indicate a better user experience. 'Meaningfulness' assesses the significance of discovered activity classes to users, an aspect not covered by the baseline. Measurements marked with an asterisk (*) indicate statistically significant ($p < 0.05$) differences between the two systems.

that the optimal prompt frequency differs by user and should also be personalized in future studies. Regarding the baseline (manual activity logging), users attributed mental burden and time pressure primarily to having to remember to start and stop recording each activity. This further justified the design of the collaborative labeling process.

Finally, our system discovered 5.63 out of 5.88 activity types on average. In the two rare instances where ActivitySeeker failed to discover an activity, the user performed the activity for less than 1 minute, too short for ActivitySeeker to establish a motion pattern. We suspect that, with a few minutes of additional data, ActivitySeeker would be able to discover and learn to recognize the activities in those 2 instances.

5.3 ActivitySeeker vs Apple Watch

5.3.1 Experiment Design. Despite its lack of ability to discover custom user-defined activities, the Apple Watch can detect nine activities (indoor walking, outdoor walking, indoor running, outdoor running, outdoor cycling, pool swim, open water swim, rowing, and elliptical), and alert the user when they start one of the activities. By comparing the user's evaluation of the meaningfulness of the activities discovered by ActivitySeeker to that of the activities detected by the Apple Watch, we can demonstrate the value of recognizing custom, user-defined activities. Moreover, both ActivitySeeker and the Apple Watch prompt the user for input from time to time. By comparing subjective user experience metrics, we can show that the prompts from ActivitySeeker do not annoy the user more than those from the Apple Watch. This in turn can demonstrate that ActivitySeeker is not too intrusive or annoying for the average user, since the Apple Watch generally delivers a good user experience.

To compare the user experience of ActivitySeeker and the Apple Watch, we designed an experiment where 6 users (4 males, 2 females, aged 21-25) installed ActivitySeeker on their smartphones while also wearing an Apple Watch. The experiment lasted 3 days, during which the users compared the experience of using ActivitySeeker and the Apple Watch to log their activities in daily life. Both ActivitySeeker and the Apple Watch can reliably detect, recognize and log activities, removing the need for an observer and allowing us to run a longer in-the-wild experiment. See Appendix F for details on the activities encountered in this experiment.

Similar to the first field study, we administered a questionnaire at the end of the experiment. In the questionnaire, apart from the metrics described in Figure 7, we also measured the intrusiveness, perceived activity discovery ability, perceived activity recognition ability, annoyance caused by interaction prompts, and the number of activities discovered. Note that the Apple Watch cannot actually discover novel activities, and the perceived activity discovery capability is caused by the Apple Watch detecting one of nine predefined types of workout and alerting the user to start recording.

5.3.2 Results. The results from this study are shown in Figure 8. We conducted statistical analysis using one-way ANOVA and reported the results in the figure. Additional results from this experiment can be found in Appendix G. The collaborative labeling process of ActivitySeeker poses a level of intrusiveness similar to that of the Apple Watch. However, ActivitySeeker has significantly better perceived activity discovery and recognition capabilities. This can be attributed to its ability to discover custom user-defined activities. Finally, the custom activities discovered by ActivitySeeker are slightly more meaningful than the nine pre-defined activities the Apple Watch is designed to discover, though statistical significance is not reached ($p = 0.065$). In conclusion, ActivitySeeker and the Apple Watch have comparable user experience, but our system has much better activity discovery and recognition capabilities.

5.4 Suitability for Deployment

The field studies provided an opportunity to evaluate the system's performance on smartphones and its suitability for real-world deployment. Specifically, we focused on inference time, training time, storage, privacy, and energy efficiency. Our results confirm that

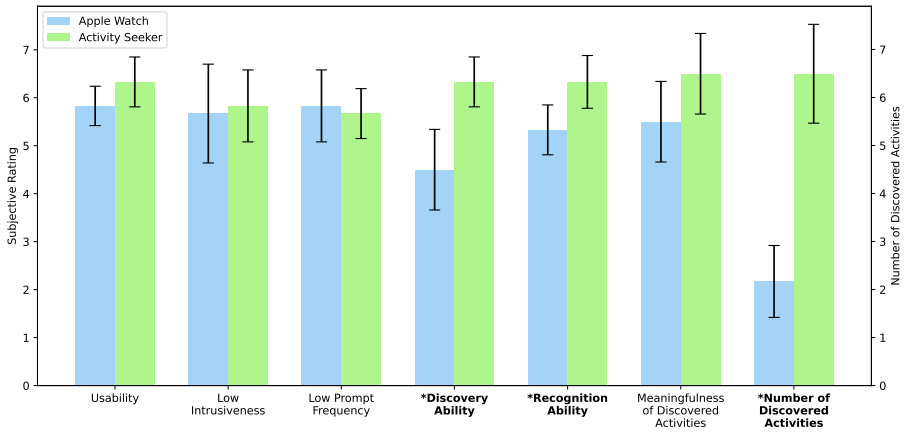


Figure 8: Users' subjective perception of ActivitySeeker and the Apple Watch, and the number of activities discovered during the three-day experiment. Significant ($p < 0.05$) differences are marked with an asterisk (*). ActivitySeeker and the Apple Watch delivered similar levels of user experience, but ActivitySeeker had substantially better activity discovery and recognition abilities.

the system is fast, energy efficient, storage-friendly and privacy-preserving:

- **Inference Time.** The system's average inference time on a smartphone is 36.3ms, which supports real-time activity recognition given that an input is received every 5 seconds.
- **On-Device Training.** Transfer learning enables fast and efficient on-device training, as shown in Table 7. On an Android smartphone, an SVM classifier can be trained with 2000 samples in 1.2 seconds, fast enough for deployment and actual use.
- **Storage.** ActivitySeeker only stores 128-dimensional feature vectors, as opposed to the original 6×500 IMU data, resulting in a substantial reduction in storage requirements. Storing thousands of these feature vectors requires just a few megabytes, negligible for modern smartphones.
- **Privacy.** All data is stored and processed on-device as feature vectors, which are much less interpretable than raw IMU data in the event of a data breach. Therefore, ActivitySeeker introduces minimal privacy risks.
- **Energy Efficiency.** We measured the power usage of an Android smartphone (4200mAh battery) under three scenarios: (1) no background processes, (2) background process collecting IMU data at 100 Hz, and (3) running the ActivitySeeker application. We kept the smartphone running under these conditions for 1 day each, and found that the average battery drain per hour of the three cases is 189 mAh, 252 mAh, and 268 mAh, respectively. These results show a limited increase in power consumption compared to the idle state, and indicate that the primary source of additional power consumption is the IMU itself, not the ActivitySeeker pipeline. Strategically lowering the IMU sampling rate during periods of inactivity could significantly improve the energy efficiency of ActivitySeeker.

In summary, our system combines real-time inference, efficient on-device training, minimal storage usage, privacy protection in case of data breach, and low power consumption, making it suitable for deployment on mobile devices.

Model	Device	Training Time (s)
SVM + Transfer Learning	Android Smartphone	1.2
SVM + Transfer Learning	Laptop	< 0.1
Full Model	Laptop	≈ 100

Table 7: Comparison of On-device training time. The training time for the whole model on Android smartphones is too long to measure. The laptop has an RTX 2080 MaxQ GPU onboard, which is used to train the full model.

6 Discussion

6.1 Applications

Our proposed system shows promise in several practical application scenarios, with the ability to accurately track a wide range of user activities. Compared to existing solutions, ActivitySeeker can automatically recognize and log custom user-defined activities, allowing for finer-grained recording of exercise sessions while causing minimal additional user burden. Moreover, combined with other data, such as time and location, the activity logs can be used for context aware recommendation systems, generating content suitable for the user's current activity.

ActivitySeeker also shows potential for crowdsourcing labeled IMU data using smartphones. With ActivitySeeker, users only need to respond to prompts from ActivitySeeker occasionally, which is much less intrusive and demanding than manual labeling. This enables the creation of much larger labeled IMU datasets that are crucial to data-centric AI research [76].

Since our system builds personalized activity recognition models from scratch and can recognize user-defined custom activities, it is suitable for users with unique motion patterns, including those with disabilities. ActivitySeeker has the potential to make accurate and personalized activity recognition more accessible, thereby improving the accessibility of other services that depend on HAR.

6.2 Limitations and Future Work

6.2.1 Public Datasets and Reproducibility. We did not use public datasets, such as UCI-HAR [7], PAMAP2 [57], Opportunity [59], and MMAct [37], in our quantitative experiments, a limitation in terms of reproducibility. This decision, as elaborated in sections 3 and 4, is primarily due to the limited data volume per user per activity offered by these datasets and the controlled environments. This also points to the lack of public datasets with more depth (i.e. more data per user). As a remedy, **we are releasing the 112.8 hours of real-world smartphone IMU data collected for this study** to improve reproducibility and complement existing datasets that focus more on breadth (i.e. covering more users). The data is available here.

6.2.2 User Experience and Long-term User Study. Although ActivitySeeker delivered a similar user experience to the Apple Watch, the user interaction process in our system could be further refined to make the collaborative labeling process less intrusive. This is especially important, considering that one user reported some level of anxiety rooted in anticipating when ActivitySeeker would ask for a label. Active learning techniques, which query the user to label only samples with low confidence, have demonstrated potential in previous personalized HAR studies [42, 45, 67]. By reducing the frequency of user interactions, active learning can lower the intrusiveness of ActivitySeeker. We also acknowledge the lack of a long-term, large-scale user study. The longest user study lasted 3 days, which is enough to show ActivitySeeker's strengths compared to the Apple Watch. However, a longer experiment lasting several weeks may offer more insights into ActivitySeeker's long term performance and benefits to users. We aim to investigate the long term impact of ActivitySeeker on users in the future.

6.2.3 Recognizing Complex Activities. In its current form, ActivitySeeker is designed to discover and recognize atomic activities. This is already useful for applications in fitness, data crowdsourcing, and accessibility, all of which benefit from customizable atomic activity recognition. However, in many downstream applications, the ability to recognize complex activities (i.e. activities with a longer duration made up of multiple atomic activities) is needed. This calls for changes to all three components of ActivitySeeker (feature extraction, collaborative labeling, and transfer learning). To effectively model complex activities, a **two-layered approach** could be used. In this approach, we train two feature extraction models - an IMU HAR model trained to extract low-level features for atomic activity recognition, and a sequential model trained to classify complex activities from sequences of low-level atomic activities. In this approach, the user labels both atomic and complex activities through collaborative labeling. The pipeline would then build a personalized atomic activity recognition model that is somewhat analogous to the *tokenizer* in NLP, as well as a sequential model for complex activity recognition using these "tokens" as input through transfer learning. A more ambitious approach would be to train a feature extractor for complex activities **end-to-end**. In this case, the ActivitySeeker pipeline can be used as-is. While this seems like a cleaner approach, it remains unclear whether it is possible to train high performance end-to-end feature extractors that are lightweight enough for mobile deployment.

6.2.4 What About Static Activities? In this study, we treated static IMU segments as noise and discarded them. This was done to reduce the compute load and power usage of ActivitySeeker. However, we acknowledge that valuable information can be mined from static IMU segments as well. For example, some smartphone IMU datasets (e.g. MotionSense [41], UCI-HAR [6], HAPT [58]) require HAR models to differentiate static activities like standing and sitting. We further observe that the smartphone is often placed on surfaces or mounted on smartphone holders (e.g. while the user is working on their computer, while the user is driving, etc.). These potentially valuable clues about the user's activity are currently ignored by ActivitySeeker. To extend ActivitySeeker to these scenarios, we suggest including more diverse static activities in the pre-training dataset, which would enable the feature extractor to learn the feature representation of static IMU data. This approach is possible due to the fact that "static" activities still come with minute but characteristic patterns in IMU data, and prior HAR work [81] on smartphone IMU datasets is able to distinguish these activities. Finally, to recognize static activities, the noise filter needs to be redesigned or removed outright. This may increase compute load and power usage, calling for the design of scheduling strategies for steps like clustering and transfer learning.

7 Conclusion

In this paper, we have presented ActivitySeeker, a novel solution for personalized HAR and the discovery and recognition of user-defined custom activities. ActivitySeeker enables activity discovery and recognition through the following steps: noise removal, feature preprocessing, clustering, collaborative labeling, and transfer learning. We evaluated ActivitySeeker through both simulated online learning and user studies, and demonstrated its superior performance compared to various baselines, including existing approaches in the academic community and a state of the art commercial device (the Apple Watch). We also showed that ActivitySeeker can work efficiently under the resource constraints of a smartphone. In conclusion, we proposed and validated an adaptive user-centric approach to activity discovery and recognition, opening up new possibilities for future research and applications.

Acknowledgments

This work is supported by National Key R&D Program of China under grants No. 2024YFB4505500 and 2024YFB4505501, Beijing Key Lab of Networked Multimedia, Institute of Artificial Intelligence, Tsinghua University (THUAI), College of AI, Tsinghua University, and Beijing National Research Center for Information Science and Technology (BNRist).

References

[1] Rebecca Adaimi and Edison Thomaz. 2022. Lifelong Adaptive Machine Learning for Sensor-Based Human Activity Recognition Using Prototypical Networks. *Sensors* 22, 18 (2022). doi:10.3390/s22186881

[2] J.K. Aggarwal and M.S. Ryoo. 2011. Human activity analysis: A review. *ACM Comput. Surv.* 43, 3, Article 16 (April 2011), 43 pages. doi:10.1145/1922649.1922653

[3] Reza Akhavian and Amir H. Behzadan. 2015. Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. *Advanced Engineering Informatics* 29, 4 (2015), 867–877. doi:10.1016/j.aei.2015.03.001 Collective Intelligence Modeling, Analysis, and Synthesis for Innovative Engineering Decision Making Special Issue of the 1st International Conference on Civil and Building Engineering Informatics.

[4] Bandar Almaslukh, Jalal AlMuhtadi, and Abdelmonim Artoli. 2017. An effective deep autoencoder approach for online smartphone-based human activity recognition. *Int. J. Comput. Sci. Netw. Secur* 17, 4 (2017), 160–165.

[5] Sizhe An, Ganapati Bhat, Suat Gumussoy, and Umit Ogras. 2023. Transfer Learning for Human Activity Recognition Using Representational Analysis of Neural Networks. *ACM Trans. Comput. Healthcare* 4, 1, Article 5 (March 2023), 21 pages. doi:10.1145/3563948

[6] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. 2012. Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine. In *Ambient Assisted Living and Home Care*, José Bravo, Ramón Hervás, and Marcela Rodríguez (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 216–223.

[7] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. 2013. A public domain dataset for human activity recognition using smartphones.. In *Esann*, Vol. 3. 3–4.

[8] Claudio Bettini, Gabriele Civitarese, and Riccardo Presotto. 2021. Personalized semi-supervised federated learning for human activity recognition. *arXiv preprint arXiv:2104.08094* (2021).

[9] Barbara Bruno, Fulvio Mastrogiovanni, Antonio Sgorbissa, Tullio Vernazza, and Renato Zaccaria. 2013. Analysis of human behavior recognition algorithms based on acceleration data. In *2013 IEEE International Conference on Robotics and Automation*. 1602–1607. doi:10.1109/ICRA.2013.6630784

[10] Davide Buffelli and Fabio Vandin. 2021. Attention-Based Deep Learning Framework for Human Activity Recognition With User Adaptation. *IEEE Sensors Journal* 21, 12 (2021), 13474–13483. doi:10.1109/JSEN.2021.3067690

[11] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* 46, 3, Article 33 (Jan. 2014), 33 pages. doi:10.1145/2499621

[12] Berardina De Carolis, Stefano Ferilli, and Domenico Redavid. 2015. Incremental Learning of Daily Routines as Workflows in a Smart Home Environment. *ACM Trans. Interact. Intell. Syst.* 4, 4, Article 20 (Jan. 2015), 23 pages. doi:10.1145/2675063

[13] Pierluigi Casale, Oriol Pujol, and Petia Radeva. 2011. Human Activity Recognition from Accelerometer Data Using a Wearable Device. In *Pattern Recognition and Image Analysis*, Jordi Vitrià, João Miguel Sanches, and Mario Hernández (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 289–296.

[14] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. 2020. A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 39 (March 2020), 30 pages. doi:10.1145/3380985

[15] Kaixuan Chen, Lina Yao, Dalin Zhang, Xianzhi Wang, Xiaojun Chang, and Feiping Nie. 2020. A Semisupervised Recurrent Convolutional Attention Model for Human Activity Recognition. *IEEE Transactions on Neural Networks and Learning Systems* 31, 5 (2020), 1747–1756. doi:10.1109/TNNLS.2019.2927224

[16] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities. *ACM Comput. Surv.* 54, 4, Article 77 (May 2021), 40 pages. doi:10.1145/3447744

[17] Federico Cruciani, Ian Cleland, Chris Nugent, Paul McCullagh, Kåre Synnes, and Josef Hallberg. 2018. Automatic Annotation for Human Activity Recognition in Free Living Using a Smartphone. *Sensors* 18, 7 (2018). doi:10.3390/s18072203

[18] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2022), 3366–3385. doi:10.1109/TPAMI.2021.3057446

[19] Florenc Demrozi, Graziano Pravadelli, Azra Bihorac, and Parisa Rashidi. 2020. Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey. *IEEE Access* 8 (2020), 210816–210836. doi:10.1109/ACCESS.2020.3037715

[20] Matthijs Douze, Arthur Szlam, Bharath Hariharan, and Hervé Jégou. 2018. Low-Shot Learning With Large-Scale Diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Ç.Berke Erdaş, Işıl Atasoy, Koray Açıcı, and Hasan Oğul. 2016. Integrating Features for Accelerometer-based Activity Recognition. *Procedia Computer Science* 98 (2016), 522–527. doi:10.1016/j.procs.2016.09.070 The 7th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2016)/The 6th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2016)/Affiliated Workshops.

[22] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. 2021. A Unified Objective for Novel Class Discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9284–9292.

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

[24] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2023. Investigating Enhancements to Contrastive Predictive Coding for Human Activity Recognition. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 232–241. doi:10.1109/PERCOM56429.2023.10099197

[25] Haibo He, Sheng Chen, Kang Li, and Xin Xu. 2011. Incremental Learning From Stream Data. *IEEE Transactions on Neural Networks* 22, 12 (2011), 1901–1914. doi:10.1109/TNN.2011.2171713

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[27] Shruthi K. Hiremath, Yasutaka Nishimura, Sonia Chernova, and Thomas Plötz. 2022. Bootstrapping Human Activity Recognition Systems for Smart Homes from Scratch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 119 (Sept. 2022), 27 pages. doi:10.1145/3550294

[28] Chunyu Hu, Yiqiang Chen, Lisha Hu, and Xiaohui Peng. 2018. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognition* 78 (2018), 277–290. doi:10.1016/j.patcog.2018.01.025

[29] Zawar Hussain, Quan Z. Sheng, and Wei Emma Zhang. 2020. A review and categorization of techniques on device-free human activity recognition. *Journal of Network and Computer Applications* 167 (2020), 102738. doi:10.1016/j.jnca.2020.102738

[30] Andrey Ignatov. 2018. Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Applied Soft Computing* 62 (2018), 915–922. doi:10.1016/j.asoc.2017.09.027

[31] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. ColloSSL: Collaborative Self-Supervised Learning for Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 17 (March 2022), 28 pages. doi:10.1145/3517246

[32] Wenchao Jiang and Zhaozheng Yin. 2015. Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In *Proceedings of the 23rd ACM International Conference on Multimedia* (Brisbane, Australia) *(MM '15)*. Association for Computing Machinery, New York, NY, USA, 1307–1310. doi:10.1145/2733373.2806333

[33] Sahak Kaghyan and Hakob Sarukhanyan. 2012. Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer. *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria* 1 (2012), 146–156.

[34] Alireza Keshavarzian, Saeed Sharifian, and Sanaz Seyedin. 2019. Modified deep residual network architecture deployed on serverless framework of IoT platform based on human activity recognition application. *Future Generation Computer Systems* 101 (2019), 14–28. doi:10.1016/j.future.2019.06.009

[35] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. 2021. Contrastive Self-supervised Learning for Sensor-based Human Activity Recognition. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*. 1–8. doi:10.1109/IJCB52358.2021.9484410

[36] Aftab Khan, Nils Hammerla, Sebastian Mellor, and Thomas Plötz. 2016. Optimising sampling rates for accelerometer-based human activity recognition. *Pattern Recognition Letters* 73 (2016), 33–40. doi:10.1016/j.patrec.2016.01.001

[37] Quan Kong, Ziming Wu, Ziwei Deng, Martin Klinkigt, Bin Tong, and Tomokazu Murakami. 2019. MMAct: A Large-Scale Dataset for Cross Modal Human Action Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[38] Nicholas D. Lane, Sourav Bhattacharya, Akhil Mathur, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. 2017. Squeezing Deep Learning into Mobile and Embedded Devices. *IEEE Pervasive Computing* 16, 3 (2017), 82–88. doi:10.1109/MPRV.2017.2940968

[39] Ching-Yi Lin and Radu Marculescu. 2020. Model Personalization for Human Activity Recognition. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 1–7. doi:10.1109/PerComWorkshops48775.2020.9156229

[40] Li Liu, Yuxin Peng, Shu Wang, Ming Liu, and Zigang Huang. 2016. Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors. *Information Sciences* 340-341 (2016), 41–57. doi:10.1016/j.ins.2016.01.020

[41] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. 2019. Mobile sensor data anonymization. In *Proceedings of the International Conference on Internet of Things Design and Implementation* (Montreal, Quebec, Canada) *(IoTDI '19)*. Association for Computing Machinery, New York, NY, USA,

49–58. doi:10.1145/3302505.3310068

[42] Andrea Mannini and Stephen S. Intille. 2019. Classifier Personalization for Activity Recognition Using Wrist Accelerometers. *IEEE Journal of Biomedical and Health Informatics* 23, 4 (2019), 1585–1594. doi:10.1109/JBHI.2018.2869779

[43] K. G. Manosha Chathuramali and Ranga Rodrigo. 2012. Faster human activity recognition with SVM. In *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*. 197–203. doi:10.1109/ICTer.2012.6421415

[44] Alan Mazankiewicz, Klemens Böhm, and Mario Berges. 2020. Incremental Real-Time Personalization in Human Activity Recognition Using Domain Adaptive Batch Normalization. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 144 (Dec. 2020), 20 pages. doi:10.1145/3432230

[45] Tudor Miu, Paolo Missier, and Thomas Plötz. 2015. Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. 1138–1147. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.170

[46] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers* (Heidelberg, Germany) *(ISWC '16)*. Association for Computing Machinery, New York, NY, USA, 92–99. doi:10.1145/2971763.2971764

[47] Qin Ni, Ian Cleland, Chris Nugent, Ana Belén García Hernando, and Iván Pau de la Cruz. 2019. Design and assessment of the data analysis process for a wrist-worn smart object to detect atomic activities in the smart home. *Pervasive and Mobile Computing* 56 (2019), 57–70. doi:10.1016/j.pmcj.2019.03.006

[48] Riku-Pekka Nikula, Konsta Karioja, Mika Pylvänäinen, and Kauko Leiviskä. 2020. Automation of low-speed bearing fault diagnosis based on autocorrelation of time domain features. *Mechanical Systems and Signal Processing* 138 (2020), 106572. doi:10.1016/j.ymssp.2019.106572

[49] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, and Uzoma Rita Alo. 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications* 105 (2018), 233–261. doi:10.1016/j.eswa.2018.03.056

[50] Preksha Pareek and Ankit Thakkar. 2021. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review* 54, 3 (2021), 2259–2322.

[51] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 74 (July 2018), 16 pages. doi:10.1145/3214277

[52] Tomas Pfister, James Charles, and Andrew Zisserman. 2014. Domain-Adaptive Discriminative One-Shot Learning of Gestures. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 814–829.

[53] Thomas Plötz, Nils Y Hammerla, and Patrick Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 1729.

[54] Thomas Plötz and Yu Guan. 2018. Deep Learning for Human Activity Recognition in Mobile Computing. *Computer* 51, 5 (2018), 50–59. doi:10.1109/MC.2018.2381112

[55] Wen Qi, Hang Su, and Andrea Aliverti. 2020. A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities. *IEEE Transactions on Human-Machine Systems* 50, 5 (2020), 414–423. doi:10.1109/THMS.2020.2984181

[56] Wen Qi, Ning Wang, Hang Su, and Andrea Aliverti. 2022. DCNN based human activity recognition framework with depth vision guiding. *Neurocomputing* 486 (2022), 261–271. doi:10.1016/j.neucom.2021.11.044

[57] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*. 108–109. doi:10.1109/ISWC.2012.13

[58] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171 (2016), 754–767. doi:10.1016/j.neucom.2015.07.085

[59] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 233–240. doi:10.1109/INSS.2010.5573462

[60] Seyed Ali Rokni, Marjan Nourollahi, and Hassan Ghasemzadeh. 2018. Personalized Human Activity Recognition Using Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). doi:10.1609/aaai.v32i1.12185

[61] Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. 2022. Class-Incremental Novel Class Discovery. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 317–333.

[62] Ramyar Saeedi, Keyvan Sasani, Skyler Norgaard, and Assefaw H. Gebremedhin. 2018. Personalized Human Activity Recognition using Wearables: A Manifold Learning-based Knowledge Transfer. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 1193–1196. doi:10.1109/EMBC.2018.8512533

[63] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller (Eds.), Vol. 12. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1999/file/8725fb777f25776ffa9076e44fcfd776-Paper.pdf

[64] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. 2000. New Support Vector Algorithms. *Neural Computation* 12, 5 (2000), 1207–1245. doi:10.1162/089976600300015565

[65] Elnaz Soleimani and Ehsan Nazerfard. 2021. Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing* 426 (2021), 26–34. doi:10.1016/j.neucom.2020.10.056

[66] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (Seoul, South Korea) *(SenSys '15)*. Association for Computing Machinery, New York, NY, USA, 127–140. doi:10.1145/2809695.2809718

[67] Timo Sztyler and Heiner Stuckenschmidt. 2017. Online personalization of cross-subjects based activity recognition models on wearable devices. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 180–189. doi:10.1109/PERCOM.2017.7917864

[68] Catherine Tong, Jinchen Ge, and Nicholas D. Lane. 2022. Zero-Shot Learning for IMU-Based Activity Recognition Using Video Embeddings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 180 (Dec. 2022), 23 pages. doi:10.1145/3494995

[69] Catherine Tong, Shyam A. Tailor, and Nicholas D. Lane. 2020. Are Accelerometers for Activity Recognition a Dead-end?. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications* (Austin, TX, USA) *(HotMobile '20)*. Association for Computing Machinery, New York, NY, USA, 39–44. doi:10.1145/3376897.3377867

[70] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2019), 3–11. doi:10.1016/j.patrec.2018.02.010 Deep Learning for Pattern Recognition.

[71] Shuangquan Wang and Gang Zhou. 2015. A review on radio based activity recognition. *Digital Communications and Networks* 1, 1 (2015), 20–29. doi:10.1016/j.dcan.2015.02.006

[72] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* 53, 3, Article 63 (June 2020), 34 pages. doi:10.1145/3386252

[73] Yanwen Wang and Yuanqing Zheng. 2018. Modeling RFID Signal Reflection for Contact-free Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 193 (Dec. 2018), 22 pages. doi:10.1145/3287071

[74] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 9.

[75] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2021. Time Series Data Augmentation for Deep Learning: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4653–4660. doi:10.24963/ijcai.2021/631 Survey Track.

[76] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: a data-centric AI perspective. *The VLDB Journal* 32, 4 (Jan. 2023), 791–813. doi:10.1007/s00778-022-00775-9

[77] Jason Wu, Chris Harrison, Jeffrey P. Bigham, and Gierad Laput. 2020. Automated Class Discovery and One-Shot Interactions for Acoustic Activity Recognition. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3313831.3376875

[78] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, Tu Nguyen, Heriberto Nieto, Scott E Hudson, Charlie Maalouf, Jax Seyed Mousavi, and Gierad Laput. 2022. Enabling Hand Gesture Customization on Wrist-Worn Devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 496, 19 pages. doi:10.1145/3491102.3501904

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

[80] Ming Zeng, Tong Yu, Xiao Wang, Le T. Nguyen, Ole J. Mengshoel, and Ian Lane. 2017. Semi-supervised convolutional neural networks for human activity recognition. In *2017 IEEE International Conference on Big Data (Big Data)*. 522–529.

doi:10.1109/BigData.2017.8257967
[81] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A Framework for Human Activity Recognition Using IMU Fusion and ConvTransformer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 88 (July 2022), 26 pages. doi:10.1145/3534584
[82] Yuanqi Zheng, Chin-Long Lee, Jia Guo, Renjie Shen, Feifei Sun, Jiaqi Yang, and Alejandro Saenz Calad. 2025. Improving EFDD with Neural Networks in Damping Identification for Structural Health Monitoring. *Sensors* 25, 22 (2025). doi:10.3390/s25226929
[83] Yexu Zhou, Tobias King, Haibin Zhao, Yiran Huang, Till Riedel, and Michael Beigl. 2024. MLP-HAR: Boosting Performance and Efficiency of HAR Models on Edge Devices with Purely Fully Connected Layers. In *Proceedings of the 2024 ACM International Symposium on Wearable Computers* (Melbourne VIC, Australia) *(ISWC '24)*. Association for Computing Machinery, New York, NY, USA, 133–139. doi:10.1145/3675095.3676624
[84] Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. 2022. TinyHAR: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers* (Cambridge, United Kingdom) *(ISWC '22)*. Association for Computing Machinery, New York, NY, USA, 89–93. doi:10.1145/3544794.3558467
[85] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* 109, 1 (2021), 43–76. doi:10.1109/JPROC.2020.3004555

A Noise Removal

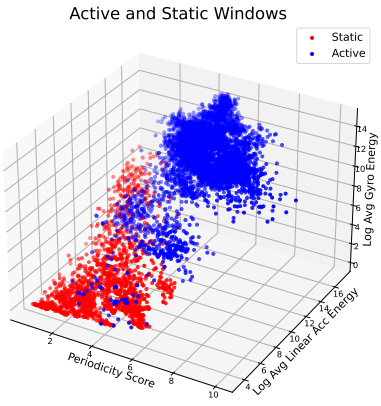


Figure 9: Distribution of active and static windows in 3-D feature space. In general, active windows are high energy and highly periodic, while static windows are low energy and less periodic.

In ActivitySeeker, a decision tree is used to decide whether a window is static or active based on 3 statistical features - the periodicity score, the log energy of the linear acceleration inputs, and the log energy of the gyroscope inputs. The periodicity score is the ratio between the maximum magnitude and the 75th percentile magnitude of the FFT spectrum [48, 82]. Intuitively, it serves as an indicator of the "peakiness" of the spectrum and a simple measure of periodicity. The effectiveness of this approach can be illustrated by visualizing active and static windows in the 3-D feature space (Figure 9). In our study, we trained a decision tree on the pre-training dataset, which contains recordings of static windows where the phone is resting on a surface, being held and used by the user, and resting in the user's pocket while the user is standing or sitting. This decision tree is then used off-the-shelf in the simulated online learning experiment and the two user studies. The results of these experiments show that the task of distinguishing between active and static windows stays roughly the same even when crossing from one user to another.

B Data Augmentation for Pre-training

Two types of data augmentation were used to train the IMU feature extraction model: scaling and time warping. This increased the amount of training data to 4 times the original amount.

For scaling, the signal is multiplied by a scaling factor sampled from a normal distribution $\mathcal{N}(1.0, 0.2)$.

For time warping, we resample the IMU signal on a warped time axis. Formally, given a signal with N time steps, we first introduce a normalized time variable $u \in [0, 1)$ with grid points $u_n = \frac{n}{N}$. We then define 4 fixed knot abscissae

$$k_x = (0, \frac{1}{3}, \frac{2}{3}, 1)$$

and construct corresponding random knot ordinates

$$k_y = (1, 1 + \epsilon_1, 1 + \epsilon_2, 1)$$

where $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, 0.1)$ are i.i.d. Gaussian perturbations controlling the strength of the time distortion. A periodic cubic spline $s(u)$ is then fitted through the control points $(k_x^{(j)}, k_y^{(j)})$, $j = 1, 2, 3, 4$, yielding a smooth modulation function $s : [0, 1] \to \mathbb{R}$. The random time warping function is defined on the normalized grid by

$$\tau(u) = Nus(u)$$

so that for each discrete index n, the warped sampling position is

$$\tau_n = \tau(u_n) = ns(\frac{n}{N}).$$

The original sequence is then resampled at these (generally non-integer) positions using linear interpolation along the discrete time axis. This procedure yields a warped sequence of the same length with local expansions and contractions in time, effectively simulating natural variations in movement speed and patterns.

C Effect of Pre-Training Dataset Composition

The composition of the pre-training dataset has deep implications on the performance of the pre-trained embedding model, and can profoundly impact the capabilities of ActivitySeeker. We tested ActivitySeeker in the simulated online learning experiment using two versions of embedding model, one trained exclusively on the MotionSense dataset and one trained on both MotionSense and data we collected. The results are shown in Table 8. By adding data from free living users that adequately reflect the within-user variance, we were able to significantly improve the performance of ActivitySeeker in the simulated online learning experiment.

Pre-training Dataset	Accuracy	F1-Score	C
MotionSense Only	0.873	0.824	7.9/8.7
MotionSense + Additional Data From 4 Users	**0.933**	**0.895**	**8.3/8.7**

Table 8: The effect of pre-training data on the performance of ActivitySeeker.

D Real-world Dataset Collection Procedure

We present the data collection procedure for both the pre-training and the real-world simulated online learning datasets.

The dataset is collected from 13 free-living participants on a university campus. All participants used their personal Android phones for the experiment. As a result, data were collected from 13 different smartphone models, capturing the diversity of devices. A data collection application was installed, enabling the participant to label the start and end of an activity event and record a labeled data segment. Participants were allowed to hold or place their phones according to their preferences without constraints on the phones' position or orientation. The data collection process lasted 8 to 20 days, with the exact duration depending on the schedule and activity level of each participant.

We held a 30-minute orientation session before data collection to familiarize the participants with the experiment settings. In the orientation session, we asked participants to collect *atomic activities* only. As defined in previous studies [2, 40, 47, 51], *atomic activities* are fundamental activities that cannot be further subdivided. For instance, complex sports like playing football or frisbee are not considered atomic, as they can be broken down into more basic activities such as running, walking, and jumping. We then asked participants to recall their daily routines and choose at least five activities, with an emphasis on diversity. Finally, we encouraged the participants to perform their chosen activities at their convenience. We emphasized that the activities should be performed naturally, and the phone should be carried in accordance with the participants' daily habits, ensuring that the data was collected from free-living environments and reflected the wearing diversity of real-life scenarios. Finally, participants were asked to discard records if they forgot to end the recording right after they finished the activity.

E Baseline User Interface

Figure 10 shows the UI of the manual labeling baseline and the Apple Watch. For the manual baseline, we included an *abort* button so that the user can discard a recording when they forgot to stop it in time to avoid contaminating the dataset. The GUI on the right is an example of the Apple Watch's *auto workout detection* function, which resembles ActivitySeeker's interaction process (i.e. prompting the user to confirm the activity label).

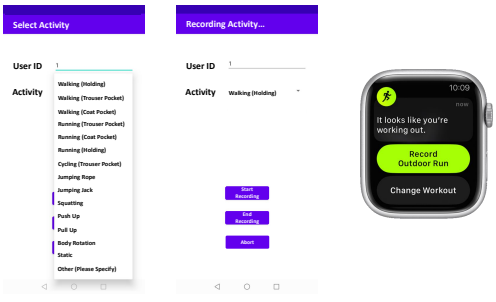


Figure 10: The UI of the two baselines - the two screenshots on the left show how users select an activity, start, and stop recordings in the manual labeling baseline.

F Activities Encountered in the User Studies

We report the activities chosen by the participants in the two user studies in Tables 9 and 10. Note that the users also performed other types of activities as they rested during the first study (ActivitySeeker vs Manual Labeling), or as they went on with their daily life in the second study (ActivitySeeker vs Apple Watch). For the second study, some of these activities (e.g. walking, cycling, running) were also recognized by the Apple Watch, while others (e.g. ascending and descending stairs, jumping jacks, etc.) are not recognized by the Apple Watch. Since the first study is mostly about the user experience of collaborative labeling and lasted only 30 minutes, the activities are relatively simple. In contrast, the second study, where the user compared the HAR capability and user experience of ActivitySeeker and the Apple Watch in-the-wild, more diverse activities were encountered.

User	Activities Chosen
1	Walking (Trouser Pocket), Walking (Holding), Walking (Coat Pocket), Running (Holding), Jumping Rope
2	Walking (Trouser Pocket), Walking (Holding), Running (Holding), Running (Trouser Pocket), Going Upstairs, Going Downstairs, Jumping Rope
3	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Jumping Rope, Going Upstairs
4	Walking (Trouser Pocket), Walking (Holding), Going Upstairs, Going Downstairs, Body Rotation (Trouser Pocket)
5	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Running (Trouser Pocket), Running (Holding), Body Rotation (Holding)
6	Walking (Holding), Walking (Swinging Arms), Running (Holding), Body Rotation (Holding), Jumping Rope, Jumping Jack
7	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Running (Trouser Pocket), Running (Holding)
8	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Running (Trouser Pocket), Running (Holding), Jumping Rope, Jumping Jack

Table 9: The activities chosen by the 8 users in the first user study, where they compared ActivitySeeker to the manual labeling baseline.

User	Discovered by Activity Seeker	Recognized by Apple Watch
1	Walking (Coat Pocket), Walking (Holding), Push Up, Running (Holding), Running (Coat Pocket), Cycling	Outdoor Run, Outdoor Cycling
2	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Running (Holding), Cycling	Outdoor Walk, Outdoor Run, Outdoor Cycling
3	Walking (Trouser Pocket), Walking (Holding), Walking (Swinging Arms), Running (Holding), Running (Trouser Pocket), Body Rotation (Holding), Jumping Rope, Jumping Jack	Outdoor Walk, Outdoor Run
4	Walking (Trouser Pocket), Walking (Holding), Going Upstairs, Jumping Rope, Cycling, Squatting	Outdoor Cycling
5	Walking (Trouser Pocket), Walking (Holding), Running (Holding), Running (Trouser Pocket), Going Upstairs, Cycling	Outdoor Run, Outdoor Cycling
6	Walking (Trouser Pocket), Walking (Swinging Arms), Running (Holding), Jumping Rope, Jumping Jack, Cycling	Outdoor Walk, Outdoor Run, Outdoor Cycling

Table 10: The activities encountered in the second user study, where they compared ActivitySeeker to the Apple Watch.

G ActivitySeeker vs Apple Watch: More Metrics

In the second field study comparing ActivitySeeker to the Apple Watch, we measured a wide range of user experience metrics. In addition to those shown in Figure 8, we also measured the physical burden, mental burden, time pressure, frustration and user preference of both ActivitySeeker and the Apple Watch. The results

are shown in Figure 11. One-way ANOVA revealed no statistically significant difference between ActivitySeeker and the Apple Watch in these 5 metrics, suggesting that the interaction experience of ActivitySeeker is user friendly, similar to that of the Apple Watch.

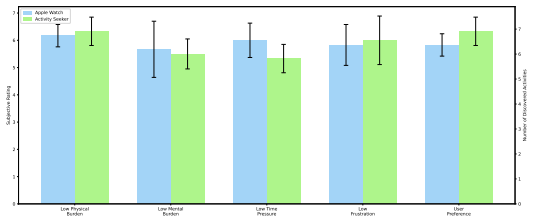


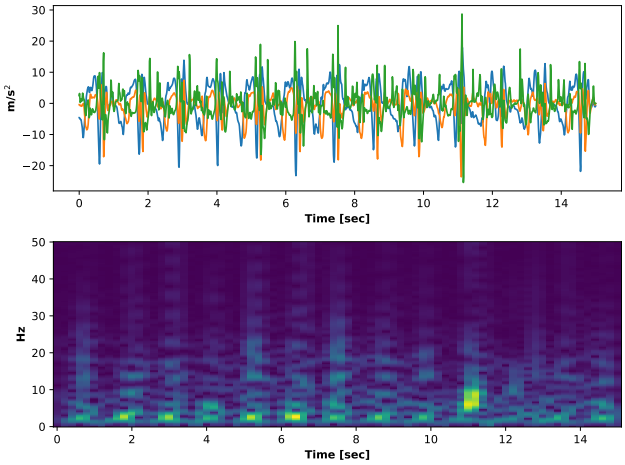
Figure 11: Users' subjective perception of ActivitySeeker and the Apple Watch. These results show no statistically significant difference between the perceived interaction experience of ActivitySeeker and the Apple Watch at the $\alpha = 0.05$ level, suggesting that ActivitySeeker is capable of delivering Apple Watch level user experience while offering significantly stronger activity discovery and recognition capabilities.

H Real-World Dataset Composition

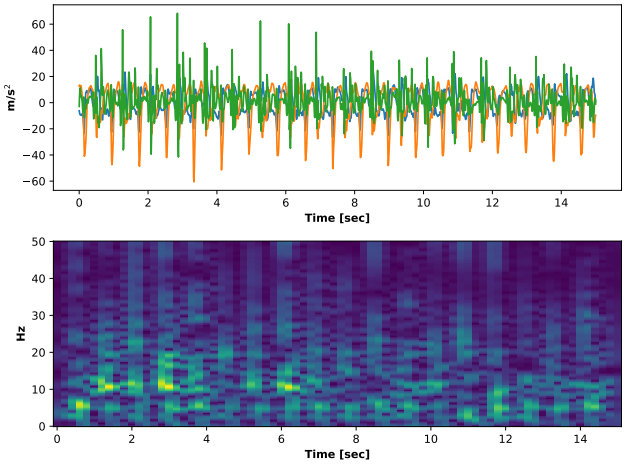
Detailed composition of the real-world dataset is shown in Table 11. It reflects the natural class imbalance of human activities, as well as the different preferences of each user. The significance of recognizing user-defined custom activities is also shown here: HAR systems rarely include "Push Up" and "Pull Up" in their pre-defined activity categories, but recognizing these workouts is useful for users 7, 8 and 9.

I Visualization of IMU Data

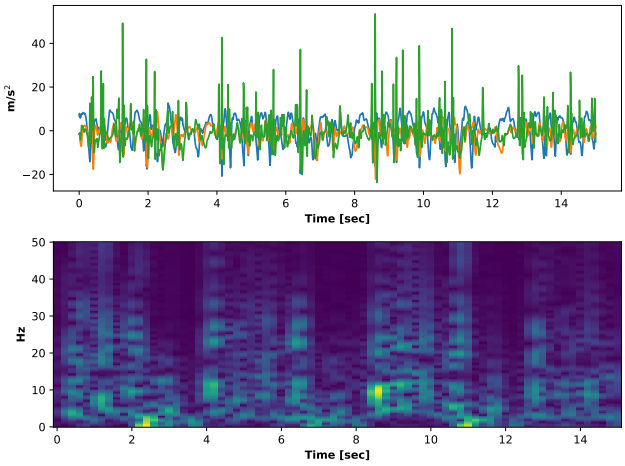
We present a visualization of raw IMU data (Figure 12). It can be observed that different activities show different characteristics in both time and frequency domains. This is the key rationale behind our integration of features from both domains. Moreover, the spectrum of periodic activities is usually dominated by a few peaks. This led to our simple and intuitive periodicity score that measures "peakiness".



(a) Walking (Trouser Pocket)



(b) Running (Trouser Pocket)



(c) Going Downstairs (Trouser Pocket)

Figure 12: Time-domain and Frequency-domain Signals of Different Activities. The time-domain signals for walking and going upstairs demonstrate similarity, whereas distinct variations in the frequency domain are observed among all four signals.

Activity	User1	User2	User3	User4	User5	User6	User7	User8	User9	User10	User11	User12	User13
Walking (Holding)	1085	2492	53	1436	792	2987	1756	967	4897	1558	1429	970	573
Walking (Coat Pocket)	1314	2265	1779	1064	795		1892	612				4011	
Walking (Trouser Pocket)	584	1960	1929	1493	1217	4156	1798			2918	2754	2539	1828
Go Upstairs (Trouser Pocket)	626	844	609			207	549	209	1153	1189	711	446	514
Go Downstairs (Trouser Pocket)	537	680	474			469	463	213	927	1115	641	377	505
Jumping Rope (Trouser Pocket)	606		499	24			350		877				
Jumping Jack (Holding)	389		445	162			431	170	610		518		
Running (Holding)	458		744	480		2017	490	720	5310	305		946	1236
Running (Coat Pocket)	456						543						
Running (Trouser Pocket)	479		880				502			305	1461		
Cycling (Trouser Pocket)	567	3475	1153	2247	2881	700	2499	1216	8948	3135	2684	2649	575
Squatting (Trouser Pocket)							428	179	1118				206
Push Up (Trouser Pocket)			161				434	97	718				
Pull Up (Trouser Pocket)							349						
Body Rotation (Trouser & Coat Pocket)	1122		597		626	38	985	232	1177		952	668	

Table 11: The composition of the real-world dataset. We report the number of 5-second windows for each activity and user. A blank means that a user did not perform an activity.