Unleashing the Power of Shared Label Structures for Human Activity Recognition

Xiyuan Zhang
University of California, San Diego
xiyuanzh@ucsd.edu

Ranak Roy Chowdhury
University of California, San Diego
rrchowdh@eng.ucsd.edu

Jiayun Zhang
University of California, San Diego
jiz069@ucsd.edu

Dezhi Hong*
Amazon
hondezhi@amazon.com

Rajesh K. Gupta
University of California, San Diego
rgupta@ucsd.edu

Jingbo Shang
University of California, San Diego
jshang@ucsd.edu

ABSTRACT

Current human activity recognition (HAR) techniques regard activity labels as integer class IDs without explicitly modeling the semantics of class labels. We observe that different activity names often have shared structures. For example, "open door" and "open fridge" both have "open" as the action; "kicking soccer ball" and "playing tennis ball" both have "ball" as the object. Such shared structures in label names can be translated to the similarity in sensory data and modeling common structures would help uncover knowledge across different activities, especially for activities with limited samples. In this paper, we propose SHARE, a HAR framework that takes into account shared structures of label names for different activities. To exploit the shared structures, SHARE comprises an encoder for extracting features from input sensory time series and a decoder for generating label names as a token sequence. We also propose three label augmentation techniques to help the model more effectively capture semantic structures across activities, including a basic token-level augmentation, and two enhanced embedding-level and sequence-level augmentations utilizing the capabilities of pre-trained models. SHARE outperforms state-of-the-art HAR models in extensive experiments on seven HAR benchmark datasets. We also evaluate in few-shot learning and label imbalance settings and observe even more significant performance gap.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; *Natural language processing*; • **Applied computing**;

KEYWORDS

human activity recognition; time series classification; label name semantics; natural language processing

ACM Reference Format:
Xiyuan Zhang, Ranak Roy Chowdhury, Jiayun Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. 2023. Unleashing the Power of Shared Label Structures for Human Activity Recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3615101

*Work unrelated to Amazon.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615101

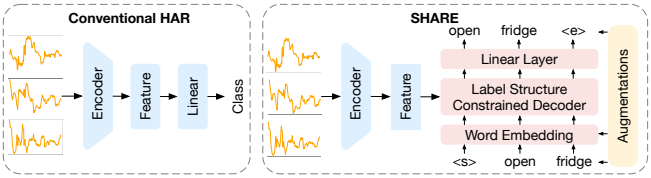


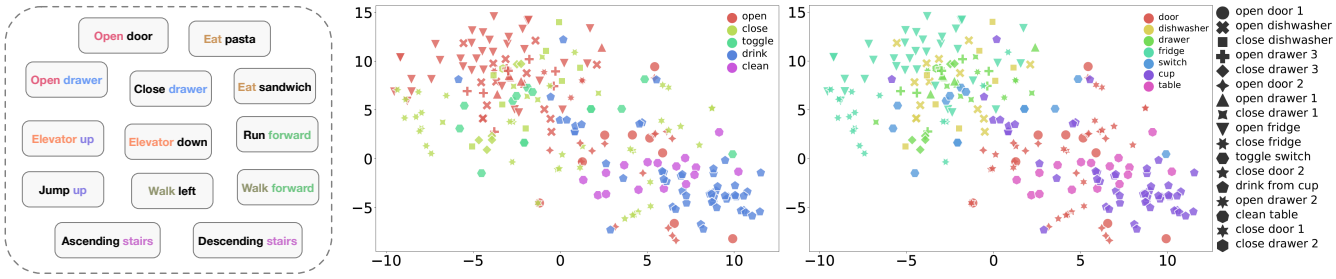
Figure 1: Existing HAR framework vs SHARE. SHARE exploits shared structures in label names and generates activity name sequences as prediction, rather than predicting integer class IDs. We also design three label augmentations at different levels to better capture shared structures.

1 INTRODUCTION

Sensor-based human activity recognition (HAR) identifies human activities using sensor readings from wearable devices. HAR has a variety of applications including healthcare, motion tracking, smart home automation, human-computer interaction [5, 8, 18, 25, 30, 50]. For example, acceleration sensors attached to legs record subjects walking around and performing daily activities for gait analysis for Parkinson's disease patients [2]; accelerometer and gyroscope can monitor user postures to detect falls for elderly people [47].

While tremendously valuable, HAR data remain difficult to collect due to security or privacy concerns, as human subjects involved in the collection process may not consent to data sharing or data transmission over the network. This often leads to local training at the edge using limited samples from just a few human subjects. Additionally, certain types of human activities happen less frequently by nature, further complicating data collection.

We note that existing HAR methods treat labels simply as integer class IDs and learn their semantics purely from annotated sensor data. This is less effective especially when labeled data are limited. To achieve better recognition performance, prior research mostly is concentrated on designing better feature extraction modules [14, 24, 33, 36] while largely overlooking the advantages of modeling label structures. Since sensory readings measuring human activities are time-series data, existing time-series classification models are also applicable to HAR. These methods, however, are also primarily focused on enhancing feature extraction [9, 12, 52]. It is noteworthy that both HAR and time-series classification methods in the literature miss the modeling of label name structures.



(a) Label

(b) T-SNE results of sensory data with the colors denoting different shared label structures.

Figure 2: (a) Labels in HAR datasets typically share common structures. (b) T-SNE visualization of sensory data in Opportunity dataset [38]. Activities with the same actions or objects (marked by the same colors) are closer. Each point represents one data sample, and each type of marker represents a different type of activity. The two figures have the same set of data points and markers and only differ in colors. The same color represents common actions (left figure) or common objects (right figure).

We argue that a more effective approach to learning activity semantics is through label name modeling, as activity names in HAR datasets often share structures that reflect the similarity between different activities. For example, both "open door" and "open fridge" (sharing the action "open") involve pulling a (fridge) door around a hinge (while "open door" first rotates the knob to release the lock and "open fridge" directly pulls the handle); both "stairs up" and "stairs down" (sharing the object "stairs") need to bend the knees and extend the legs. Figure 2a illustrates more examples of activity label names in typical HAR datasets (e.g., "eat pasta" and "eat sandwich", "elevator up" and "elevator down"). The common actions or objects in these examples translate to similarities in the IMU data space. As shown in Figure 2b, we apply t-SNE visualization on sensor readings from the Opportunity dataset [38]. We color different activities by common actions or objects. Activities of the same color (sharing the same action or object in label names) appear closer in the embedding space, indicating stronger similarity in the original sensory measurements. Such mapping between input features and label names motivates us to design a more effective learning framework that extracts knowledge from label structures.

To this end, we propose SHARE, shown in Figure 1, which models both input sensory data features and label name structures. SHARE comprises an encoder for extracting features from sensory input and a decoder for predicting label names. Unlike existing HAR models that output integer class IDs as prediction results, SHARE outputs *label name sequences*, thus preserving structures among various activities and providing a global view of activity relationships. During training, we optimize the model by minimizing the differences between predicted label names and ground-truth label names. During inference, we exploit a constrained decoding method to produce only valid labels.

We also design three label augmentation methods at different levels to better capture shared structures across activities. The basic token-level augmentation randomly replaces the original label sequences by their meaningful tokens (e.g., all actions of "eat X" are treated as a class of "eat"). This happens only during training and helps the model consolidate semantics of shared structures across different activities. We further develop two embedding- and sequence-level augmentations leveraging pre-trained models. At the embedding level, we integrate pre-trained word embeddings to capture shared semantic meanings not obvious in label names (e.g., the similarity between "walk" and "run"). At the label sequence level, for HAR datasets that do not have shared structures in their original labels, we offer an automated label generation method to generate new labels with shared tokens while preserving the same semantic meanings, leveraging large language models. Specifically, we use OpenAI's GPT-4 [32] to extend atomic, non-overlapping label names into sequences of meaningful tokens. To the best of our knowledge, SHARE is the first solution to HAR classification via decoding label sequences. We evaluate SHARE on seven HAR benchmark datasets and observe the new state-of-the-art performance. We summarize our main contributions as follows:

- We find shared structures in label names map to similarity in the input data, leading to a more effective HAR framework, SHARE, by modeling label structures. SHARE captures knowledge across activities by uncovering information from label structures.
- We propose three label augmentation methods, each targeting at a different level, to more effectively identify shared structures across activities. These include a basic token-level augmentation and two pre-trained model-enhanced augmentations at the embedding level and at the label sequence level.
- We evaluate SHARE on seven HAR benchmark datasets and observe the new state-of-the-art performance. We also conduct experiments under few-shot settings and label imbalance settings and observe even more significant performance improvement.

2 RELATED WORK

2.1 Human Activity Recognition

Existing HAR approaches can be categorized into statistical methods and deep learning-based methods [6, 56]. Traditional methods are based on data dimensionality reduction, spectral feature transformation (e.g., Fourier transformation), kernel embeddings [34], first-order logic [51] or handcrafted statistical feature extraction (e.g., mean, variance, maximum, minimum) [15]. These features are then used as input to shallow machine learning methods like SVM, and Random Forest. In recent years, deep learning methods have advanced automatic feature extraction and have begun to substitute hand-crafted feature engineering in HAR [14, 16, 50]

, including convolutional neural networks, recurrent neural network, attention mechanism, and their combinations. DeepConvLSTM [33] is composed of convolutional layers for feature extractors and recurrent layers for capturing temporal dynamics of the feature representations. MA-CNN [36] designs modality-specific architecture to first learn sensor-specific information and then unify different representations for activity recognition. SenseHAR [20] proposes a sensor fusion model that maps raw sensory readings to a virtual activity sensor, which is a shared low-dimensional latent space. AttnSense [29] further integrates attention mechanism to convolutional neural network and gated recurrent units network. THAT [24] proposes a two-stream convolution augmented Transformer model for capturing range-based patterns. We shall note that these models focus on designing more effective feature extractors for better performance but neglect the semantic information in label names, which is the focus of this work.

2.2 Time-Series Classification

HAR data are time-stamped sensory series, enabling the use of time-series classification methods. Existing time-series classification models fall into two categories: statistical methods and deep learning methods. Statistical methods are based on nearest neighbor [3, 41], dictionary classifier [39], ensemble classifier [27, 40], etc. These statistical methods are more robust to data scarcity but do not scale well when the feature numbers in high-dimensional space become huge. On the other hand, deep learning methods can extract features from high-dimensional data but require abundant data points to train an effective model.

Convolutional Networks (FCN and ResNet) [19, 45] and Recurrent Neural Networks [21, 22] show better performance compared with statistical methods. TapNet [58] is an attentional prototype network that calculates the distance to class prototypes to learn feature representations. ShapeNet [26] performs shapelet selection by embedding shapelet candidates into a unified space and trains the network with cluster-wise triplet loss. SimTSC [53] formulates time-series classification as a graph node classification problem and uses a graph neural network to model similarity information. Recently, Rocket [12] applies plenty of random convolution kernels for data transformation and attains state-of-the-art accuracy. MiniRocket [13] maintains the accuracy and improves the processing time of Rocket. TST [52] and TARNet [10] incorporate unsupervised representation learning which offers benefits over fully supervised methods on the downstream classification tasks. Similar to existing HAR methods, time-series classification models focus on designing more advanced feature extraction or unsupervised representation learning methods without taking into account the label semantics, whereas SHARE models the shared structures in the label set for more effective representation learning.

2.3 Label Semantics Modeling

Given label name semantics as prior knowledge, classification tasks could benefit from modeling such semantics through knowledge graph [43] or textual information [23, 35, 54, 59]. Tong et al. [42] exploit knowledge from video action recognition models to construct an informative semantic space that relates seen and unseen activity classes. Recent works designed specifically for zero-shot learning in human activity recognition also combine semantic embeddings [31, 44, 49]. However, these works mostly calculate the mean embeddings for labels with multiple words, which misses label structures and is suboptimal. Unlike these works, SHARE preserves label structures and enables knowledge sharing through decoding label names for generic HAR.

Algorithm 1: SHARE Framework

Input : Training set $\mathcal{D}_{\text{tr}} = \{\mathbf{x_i}, \mathbf{y_i}\}_{i=0}^{N_{\text{tr}}}$, test set $\mathcal{D}_{\text{te}} = \{\mathbf{x_i}, \mathbf{y_i}\}_{i=0}^{N_{\text{te}}}$.
Model : Encoder f_θ, Decoder g_ϕ.
Output: Predicted label sequences on test set $\{\hat{\mathbf{y}}_\mathbf{i}\}_{i=0}^{N_{\text{te}}}$.

if *no shared tokens in* $\mathcal{Y}$ **then**
  Sequence-Level augmentation to $\mathcal{Y}$; // Sec 4.4
while *not converge* **do**
  Sample $(\mathbf{x_i}, \mathbf{y_i}) \sim \mathcal{D}_{\text{tr}}$;
  Token-Level augmentation $\mathbf{y}'_\mathbf{i} \leftarrow \mathbf{y_i}$; // Sec 4.3
  Encoder feature extraction $\mathbf{z_i} = f(\mathbf{x_i}; \theta)$; // Sec 4.1
  Embedding-Level augmentation and label sequence decoding $\hat{\mathbf{y}}_\mathbf{i} = g(\mathbf{z_i}; \phi)$; // Sec 4.2,4.4
  Optimize θ and ϕ through Equation 4;
for $(\mathbf{x_i}, \mathbf{y_i}) \in \mathcal{D}_{\text{te}}$ **do**
  Encoder feature extraction $\mathbf{z_i} = f(\mathbf{x_i}; \theta)$;
  Embedding-Level augmentation and label sequence constrained decoding $\hat{\mathbf{y}}_\mathbf{i} = \text{argmax}_{\mathbf{y_i} \in \mathcal{Y}} P_\phi(\mathbf{y_i}|\mathbf{z_i})$;
return predicted label sequences $\{\hat{\mathbf{y}}_\mathbf{i}\}_{i=0}^{N_{\text{te}}}$.

3 PRELIMINARY

We focus on human activity recognition such as walking and sitting, captured by the sensory time-series data in a given time period. We formulate HAR settings of conventional methods and SHARE.

Conventional HAR. We denote HAR dataset in conventional methods as $\mathcal{D}' = \{(\mathbf{x_i}, c_i)\}_{i=0}^{N}, \mathbf{x_i} \sim \mathcal{X}, c_i \sim \mathcal{C}$, where $\mathcal{X}$ and $\mathcal{C}$ denote the input space and the label space. Each sample of time-series input is denoted as $\mathbf{x_i} \in \mathbb{R}^{T_i \times v}$, where T_i is the length of the time series, and v is the number of measured variables. The label space $\mathcal{C}$ contains C classes, and each label c is an integer from $\{1, 2, \cdots, C\}$.

SHARE. We denote dataset in SHARE as $\mathcal{D} = \{(\mathbf{x_i}, \mathbf{y_i})\}_{i=0}^{N}, \mathbf{x_i} \sim \mathcal{X}, \mathbf{y_i} \sim \mathcal{Y}$, where data space $\mathcal{X}$ is the same as conventional HAR methods, and $\mathcal{Y}$ denotes the label space in SHARE. We denote $\mathbf{y_i} = [y_{i1}, y_{i2}, \cdots, y_{ik_i}]$ as a sample human activity label sequence, where k_i is the length of the label sequence $\mathbf{y_i}$. For example, the label "walk upstairs" contains a word sequence of length two, ["walk", "upstairs"] respectively. The label space $\mathcal{Y}$ contains C classes and M tokens. Instead of presenting labels as independent integer IDs, there exist shared structures across different labels in the label space $\mathcal{Y}$. For example, "walk upstairs" and "walk downstairs" both have "walk" in label names. Formally, there exist labels $\mathbf{y_i}, \mathbf{y_j}, i \neq j$ that have the same word $y_{im} = y_{jl}$, where $1 \leq m \leq k_i, 1 \leq l \leq k_j$ are positions in $\mathbf{y_i}$ and $\mathbf{y_j}$.

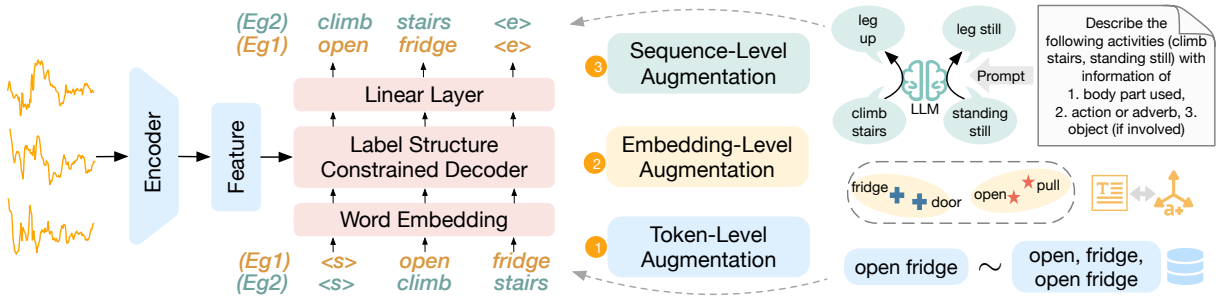


Figure 3: Framework of SHARE. We encode the time-series features and decode the label sequences as predictions. We further design three augmentation methods at different levels to better capture the shared semantic structures.

4 METHODOLOGY

We design a label structure decoding architecture for HAR, called SHARE, that exploits label structures and promotes knowledge sharing across activities. SHARE consists of two modules: Time-Series Encoder and Label Structure-Constrained Decoder. We pass multivariate sensory readings as input to the encoder and use the extracted feature vector to initialize the hidden states of the decoder. The decoder generates an activity name sequence (e.g., "climb stairs") as the prediction label. By binding sensory features with label structures, the structures in label names help the model better learn the similarity in the sensory data. We further propose three augmentation methods, including one basic token-level augmentation (randomly selecting from "climb", "stairs", "climb stairs") and two pre-trained model-enhanced augmentations at embedding (using pre-trained embeddings to initialize "climb" and "stairs" word embeddings) and label sequence levels (rephrasing "climb stairs" as "leg up" which share more tokens with other label names), to better capture shared structures across different activities. We summarize the pipeline of SHARE in Figure 3 and Algorithm 1.

4.1 Time-Series Encoder

We use $f_\theta : \mathcal{X} \to \mathcal{Z} \subset \mathbf{R}^d$ parameterized by θ to denote the time-series encoder. This part appears in both conventional HAR and SHARE. The encoder maps data from the input space $\mathcal{X}$ to the d-dimensional hidden space $\mathcal{Z}$. For conventional HAR, the final predictions are obtained from the hidden representations after a fully connected layer fc. Denote $\hat{c}_i = \text{fc}(f(\mathbf{x_i}; \theta)) \in \mathbf{R}^C$ as the distribution of the predicted label. Optimization is based on the cross-entropy loss between prediction $\hat{c}_i$ and ground truth c_i:

$$\arg\min \mathbb{E}_{(\mathbf{x_i}, c_i)\sim\mathcal{D}'} \text{CE}(c_i, \hat{c}_i). \tag{1}$$

In SHARE, the encoded representations $\mathbf{z_i} = f(\mathbf{x_i}; \theta)$ are used to initialize hidden states of the decoder, instead of being directly used for classification. This transfers learned representations from the encoder to inform the structured decoding process. To instantiate the time-series encoder, we keep both efficacy and efficiency in mind, given that HAR models usually run on edge devices with limited compute. Therefore, we use one-dimensional Convolutional Neural Networks (CNN), as they are relatively lightweight with superior capability in extracting time-series features [11, 36, 45, 57].

4.2 Label Structure-Constrained Decoder

We use $g_\phi : \mathcal{Z} \to \mathcal{Y}$ parameterized by ϕ to denote the label structure-constrained decoder in SHARE. The decoder generates word sequences in the label space $\mathcal{Y}$ given the encoded representations as initialization of the decoder hidden states. Following our notation in Section 3 (Problem Setting), we further require that each label name sequence starts from a start token $\langle s \rangle$ and ends at an ending token $\langle e \rangle$. Specifically, $\mathbf{y_i} = [y_{i0}, y_{i1}, y_{i2}, \cdots, y_{ik_i}, y_{ik_i+1}]$, where $y_{i0} = \langle s \rangle$, $y_{ik+1} = \langle e \rangle$. Decoding the token $\langle e \rangle$ means that we reach the end of the label sequence. At each decoding step, we estimate the conditional probability P_ϕ of decoding label $\mathbf{y_i}$ from $\mathbf{x_i}$, given the encoded representations $\mathbf{z_i}$ from the encoder as:

$$P_\phi(y_{i1}, y_{i2}, \cdots, y_{ik_i+1}|\mathbf{z_i}) = \prod_{t=1}^{k_i+1} P_\phi(y_{it}|\mathbf{z_i}, y_{i0}, y_{i1}, \cdots, y_{it-1}). \tag{2}$$

Training. During the training of SHARE, we adopt the teacher forcing strategy [48] where the ground truth label token y_{it} at each decoding step t is used as input to be conditioned on for predictions at decoding step $t+1$. Teacher forcing improves convergence speed and stability during training. We optimize SHARE based on cross-entropy loss between the predicted label sequence $\hat{\mathbf{y}}_\mathbf{i}$ and the ground truth label sequence $\mathbf{y_i}$:

$$\hat{\mathbf{y}}_\mathbf{i} = g(f(\mathbf{x_i}; \theta); \phi), \tag{3}$$

$$\arg\min \mathbb{E}_{(\mathbf{x_i}, \mathbf{y_i})\sim\mathcal{D}} \frac{1}{k_i} \sum_{j=1}^{k_i} \text{CE}(y_{ij}, \hat{y}_{ij}), \tag{4}$$

where $\hat{y}_{ij} \in \mathbf{R}^M$ indicates distribution of jth predicted token of $\hat{\mathbf{y}}_\mathbf{i}$.
Inference with Constrained Decoding. During inference decoding, predicted label token $\hat{y}_{it}$ from the current decoding step t is used as input to be conditioned on for predicting tokens at step $t+1$. In typical natural language processing tasks, e.g., machine translation, it is common to decode the sequence using beam search during inference. However, beam search would not work properly as it only tracks a pre-defined number of best partial solutions as candidates in decoding, and the final predictions may not belong to our label space. To guarantee that all generated labels are valid, we adopt a *constrained decoding* method. We start from the start token and iterate over all valid label sequences in the label set. We then calculate the probability of decoding each valid label sequence and choose the one with the highest probability as the final predicted label. The decoding is constrained as we only keep track of the valid partial sequences during decoding. In HAR datasets, the size of the

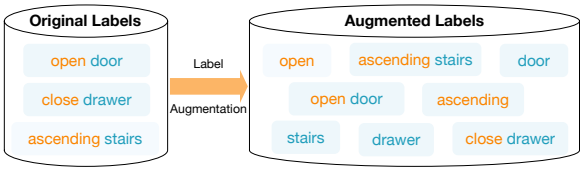


Figure 4: Illustration of basic token-level augmentation. We augment the original label name sequence by randomly choosing its meaningful tokens or the sequence itself .

label set is relatively small, and constrained decoding consumes only a small constant of memory (the size of the label set). At step t, we calculate the probability for all the valid partial sequences of length t and pass them into the decoder for generating tokens at step $t+1$. The final inference prediction is the sequence that maximizes the overall sequence probability:

$$\hat{\mathbf{y}}_\mathbf{i} = \arg\max_{\mathbf{y}_\mathbf{i} \in \mathcal{Y}} P_\phi(\mathbf{y}_\mathbf{i}|\mathbf{z}_\mathbf{i}). \tag{5}$$

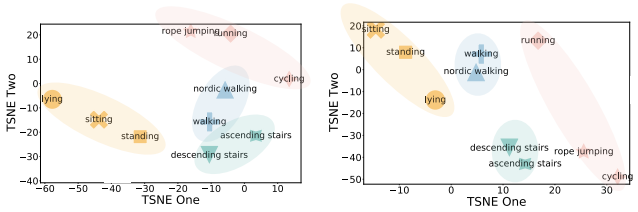
We use Long Short-Term Memory (LSTM) as an example for our label structure-constrained decoder, given its effectiveness in modeling sequential dependencies [33]. We transform the CNN-extracted features $\mathbf{z}_\mathbf{i}$ through two separate linear layers to initialize the hidden state and cell state of LSTM.

4.3 Basic Token-Level Label Augmentation

To better learn the semantics of each token in the label sequence, we apply a token-level label augmentation strategy as illustrated in Figure 4. During training, with pre-defined probability, we randomly choose meaningful single words from the original label sequence as the new labels. For example, an original label sequence "ascending stairs" contains single words "ascending" and "stairs", so we randomly select from "ascending", "stairs", and "ascending stairs" as the new labels during training. Following notation in Section 3 (Problem Setting), the original label $\mathbf{y}_\mathbf{i} = [y_{i1}, y_{i2}, \cdots, y_{ik_i}]$ is augmented as a set of new labels $\{\mathbf{y}_\mathbf{i}, y_{i1}, y_{i2}, \cdots, y_{ik_i}\}$ containing the label sequence $\mathbf{y}_\mathbf{i}$ and its meaningful tokens. For each iteration, with a pre-defined probability we randomly select the new label $\mathbf{y}'_\mathbf{i}$ from the new label set as the actual label. Optimization with token-level label augmentation can be formulated as:

$$\arg\min \mathbb{E}_{(\mathbf{x}_\mathbf{i}, \mathbf{y}_\mathbf{i}) \sim D} \mathbb{E}_{\mathbf{y}'_\mathbf{i} \sim \{\mathbf{y}_\mathbf{i}, y_{i1}, y_{i2}, \cdots, y_{ik_i}\}} \frac{1}{k'_i} \sum_{j=1}^{k'_i} \mathrm{CE}(y'_{ij}, \hat{y}_{ij}), \tag{6}$$

where k'_i is the length of the new label $\mathbf{y}'_\mathbf{i}$, y'_{ij} is the jth token of $\mathbf{y}'_\mathbf{i}$, and $\hat{y}_{ij}$ is the distribution of the predicted jth token. Since the goal of label augmentation is to help the model better capture the semantics of different activities, we only choose meaningful single tokens in the original label sequences (e.g., actions and objects) as new labels. Other single tokens like stop words or numbers (e.g., "1" in "open door 1") will not count as new labels. Note that the token-level augmentation is only applied during training. During evaluation, the ground truth label stays the same as the original label. Because we adopt a constrained decoding method during inference, it is guaranteed that all the generated label sequences are valid sequences in the original label sets.



(a) Data (b) ImageBind Embeddings

Figure 5: T-SNE visualizations show analogous clusters between input data and ImageBind word embeddings. Activities in the same color represent clusters of similar activities.

4.4 Enhanced Embedding-Level and Sequence-Level Augmentations

Apart from the basic token-level augmentation, we also develop two enhanced augmentation techniques to better capture label structures from embedding and sequence levels by leveraging the power of pre-trained models.

Embedding-Level Augmentation. Our label structure decoding architecture can capture label structures explicitly presented as shared label names. Yet, apart from these explicit shared label names, there may also exist semantic structures that implicitly span across different activities. For example, "walk" and "run" are similar activities involving the movement of legs, but they don't directly share label names. We have observed that such semantic structures can be captured by word embeddings from pre-trained models. We thus propose to use word embeddings from pre-trained models to initialize our decoder's word embedding layer, replacing the original random initialization. Specifically, we utilize word embeddings from ImageBind, a multimodal pre-trained model that learns a joint embedding space across six modalities. As shown in Figure 5, we apply t-SNE visualization to both the ImageBind word embeddings and the input sensor readings from some example activities in PAMAP dataset [37]. For activity names comprising multiple tokens, we calculate the average embedding of the aggregated tokens. T-SNE visualizations show similar clusters between ImageBind word embeddings and original data embeddings. As a result, incorporating pre-trained word embeddings helps SHARE better capture semantic structures.

Sequence-Level Augmentation. Most HAR datasets have sufficient overlapping structures in label names. However, there also exist datasets that do not have or rarely have shared tokens in their original label names. For these datasets, we can use large-scale language models to automatically generate label names with shared tokens. Specifically, we employ GPT-4 with the following prompt:

Describe the following activities one by one with information of 1. body part used, 2. action or adverb, 3. object (if involved). Please maximize the number of shared tokens across different activities and make the description as short as possible.

As human activities naturally have shared actions and objects, the prompt helps find common tokens across activities. With the aid of pre-trained language model, such a process is performed with minimal human expert effort. Based on the structured information provided by the pre-trained model, we can summarize the label names with shared tokens. We apply sequence-level augmentation mostly for datasets without original shared tokens. If the target

Table 1: Dataset statistics and an example subset of shared label names.

Dataset	Train	Test	Window Size	Channel	Class Num	An Example Subset of Shared Label Names
Opportunity [38]	2891	235	150	45	17	open door, open drawer, close drawer, open fridge, open dishwasher
PAMAP2 [37]	14438	2380	512	27	12	ascending stairs, descending stairs, walking, nordic walking
UCI-HAR [1]	7352	2947	128	9	6	walk, walk upstairs, walk downstairs
USCHAD [55]	17576	9769	100	6	12	run forward, walk forward, elevator up, elevator down, jump up
WISDM [46]	12406	3045	200	6	18	eating soup, eating pasta, kicking soccer ball, playing tennis ball
Harth [28]	14166	3588	300	6	12	sitting, standing, cycling sitting, cycling standing, cycling sitting inactive

Table 2: Accuracy and Macro-F1 for SHARE and baselines. We bold the best score and underline the second best.

Datasets	Metrics	DeepConvLSTM [33]	XGBoost [7]	MA-CNN [36]	HHAR-net [14]	TST [52]	TARNet [10]	Rocket [12]	THAT [24]	SHARE
Opp	Accuracy	0.746±0.049	0.688±0.017	0.549±0.029	0.753±0.027	0.784±0.018	0.789±0.024	0.811±0.008	0.803±0.012	**0.849±0.015**
	Macro-F1	0.634±0.036	0.547±0.011	0.416±0.036	0.620±0.021	0.668±0.023	0.669±0.034	0.670±0.016	0.691±0.015	**0.766±0.013**
PAMAP2	Accuracy	0.891±0.012	0.939±0.003	0.926±0.011	0.885±0.031	0.922±0.037	0.931±0.011	0.928±0.008	0.943±0.005	**0.960±0.002**
	Macro-F1	0.884±0.018	0.939±0.007	0.925±0.012	0.893±0.031	0.925±0.039	0.935±0.010	0.934±0.008	0.949±0.005	**0.965±0.002**
UCI-HAR	Accuracy	0.900±0.016	0.907±0.003	0.921±0.025	0.926±0.005	0.926±0.005	0.904±0.011	0.939±0.002	0.906±0.007	**0.960±0.002**
	Macro-F1	0.899±0.016	0.906±0.003	0.921±0.024	0.926±0.005	0.925±0.006	0.904±0.011	0.942±0.002	0.909±0.006	**0.959±0.002**
USCHAD	Accuracy	0.574±0.016	0.571±0.007	0.543±0.044	0.524±0.011	0.641±0.028	0.564±0.037	0.580±0.005	0.643±0.015	**0.674±0.041**
	Macro-F1	0.557±0.015	0.573±0.006	0.520±0.047	0.523±0.009	0.594±0.023	0.533±0.021	0.601±0.007	0.619±0.012	**0.627±0.027**
WISDM	Accuracy	0.689±0.014	0.668±0.005	0.634±0.059	0.566±0.016	0.715±0.003	0.733±0.011	0.643±0.007	0.774±0.005	**0.794±0.003**
	Macro-F1	0.685±0.013	0.662±0.006	0.631±0.060	0.538±0.012	0.710±0.004	0.737±0.010	0.767±0.004	0.634±0.005	**0.790±0.004**
Harth	Accuracy	0.979±0.006	0.977±0.001	0.973±0.016	0.981±0.001	0.974±0.005	0.962±0.009	0.897±0.003	0.960±0.016	**0.983±0.007**
	Macro-F1	0.578±0.032	0.522±0.003	0.538±0.025	0.515±0.049	0.501±0.031	0.481±0.031	0.472±0.019	0.485±0.025	**0.593±0.020**

HAR dataset already has sufficient overlapping tokens, we will directly use the original label names provided by human experts.

5 EVALUATION

5.1 Datasets, Baselines, and Metrics

We use six HAR benchmark datasets for evaluation, summarized in Table 1 with examples of shared label names. We split data and choose window size following previous works [10, 20]. The training and testing split is based on different participating subjects.

Opportunity[1] [38] collects readings from 4 users with 6 runs per user. Sensors include body-worn, object, and ambient sensors. The full dataset includes annotations on multiple levels, and we use mid-level gesture annotations which contain shared label structures.

PAMAP2[2] [37] comprises readings collected from 9 subjects wearing 3 IMUs sampled at 100 Hz and a heart rate monitor sampled at 9Hz. Three IMUs are positioned over the wrist on the dominant arm, on the chest, and on the dominant side's ankle, respectively.

UCI-HAR[3] [1] is collected from a group of 30 volunteers. A Samsung Galaxy S II smartphone was attached on their waist. Feature vectors were further extracted from each sliding window of the collected data in the time and frequency domain.

USCHAD[4] [55] involves 14 subjects performing 12 low-level activities. They use MotionNode (6-DOF IMU designed for human motion sensing applications) to collect the datasets.

WISDM[5] [46] is collected from accelerometer and gyroscope sensors in smartphone and smartwatch at a rate of 20Hz. 51 subjects perform 18 activities for 3 minutes respectively.

Harth[6] [28] involves 22 subjects using two three-axial accelerometers attached to the thigh and lower back, and a chest-mounted camera (for data annotation) to collect data of 12 activities.

We compare SHARE with a list of human activity recognition (DeepConvLSTM [33], MA-CNN [36], HHAR-net [14], THAT [24]) and time-series classification baselines (XGBoost [7], Rocket [12], TST [52], TARNet [10]), including both statistical approaches and state-of-the-art deep learning-based models.

We evaluate the performance of SHARE and baselines using accuracy and macro-F1. Macro-F1 is defined as macro-F1= $\frac{1}{C}\sum_{i=1}^{C} 2\times \frac{\text{Prec}_i\times\text{Rec}_i}{\text{Prec}_i+\text{Rec}_i}$, where $\text{Prec}_i, \text{Rec}_i$ represent the precision and recall for each category i, and C is the total number of categories.

5.2 Experimental Setup

We use a two-layer convolutional neural network as the encoder for extracting features. The kernel sizes for both layers are set to 3 and each layer is followed by batch normalization. We adopt LSTM with a hidden dimension of 128 as the decoder, based on a grid search of $\{64, 128, 256\}$. We use Adam optimizer with learning rate $1e^{-4}$ based on a grid search of $\{1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}\}$ and batch size 16. For all datasets, we further randomly split the training set into 80% for training and 20% for validation. We conduct the experiments in PyTorch with NVIDIA RTX A6000 (with 48GB memory), AMD EPYC 7452 32-Core Processor, and Ubuntu 18.04.5 LTS. We tune the hyper-parameters of both SHARE and baselines on the validation set and then combine training and validation set to re-train the models after hyper-parameter tuning.

5.3 Results

We repeat 5 runs and report the average accuracy, macro-F1 score, and standard deviations of SHARE and baselines in Table 2. We see that SHARE consistently outperforms both statistical and deep

[1] https://archive.ics.uci.edu/ml/datasets/opportunity+activity+recognition
[2] http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring
[3] http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones
[4] https://sipi.usc.edu/had/
[5] https://archive.ics.uci.edu/ml/datasets/WISDM+Smartphone+and+Smartwatch+Activity+and+Biometrics+Dataset+
[6] https://github.com/ntnu-ai-lab/harth-ml-experiments

Table 3: Accuracy and Macro-F1 for different model variants. We bold the best score and underline the second best.

Datasets	Metrics	VanillaHAR no label modeling	VanillaHAR+ ImageBind	multi label	no aug	no token aug	no embed aug	best baseline	SHARE
Opp	Accuracy	0.745±0.015	0.759±0.010	0.755±0.030	0.819±0.005	0.823±0.022	0.847±0.015	0.811±0.008	**0.849±0.015**
	Macro-F1	0.618±0.023	0.632±0.009	0.624±0.034	0.732±0.014	0.737±0.019	0.741±0.029	0.691±0.015	**0.766±0.013**
PAMAP2	Accuracy	0.921±0.033	0.933±0.011	0.926±0.011	0.951±0.006	0.952±0.005	0.956±0.008	0.943±0.005	**0.960±0.002**
	Macro-F1	0.924±0.024	0.938±0.014	0.928±0.010	0.960±0.006	0.958±0.005	0.964±0.009	0.949±0.005	**0.965±0.002**
UCI-HAR	Accuracy	0.921±0.005	0.928±0.005	0.926±0.005	0.954±0.006	0.957±0.004	0.958±0.004	0.939±0.002	**0.960±0.002**
	Macro-F1	0.921±0.005	0.928±0.005	0.926±0.005	0.953±0.006	0.957±0.004	0.958±0.004	0.942±0.002	**0.959±0.002**
USCHAD	Accuracy	0.543±0.028	0.566±0.013	0.550±0.023	0.622±0.050	0.635±0.034	0.663±0.012	0.643±0.015	**0.674±0.041**
	Macro-F1	0.538±0.024	0.558±0.012	0.546±0.016	0.600±0.029	0.615±0.023	0.623±0.011	0.619±0.012	**0.627±0.027**
WISDM	Accuracy	0.644±0.006	0.655±0.004	0.649±0.010	0.788±0.006	0.786±0.008	0.793±0.008	0.774±0.005	**0.794±0.003**
	Macro-F1	0.639±0.006	0.648±0.003	0.645±0.012	0.783±0.008	0.784±0.008	0.786±0.009	0.767±0.004	**0.790±0.004**
Harth	Accuracy	0.977±0.005	0.980±0.002	0.979±0.007	0.981±0.006	0.975±0.008	0.982±0.003	0.981±0.001	**0.983±0.007**
	Macro-F1	0.481±0.004	0.487±0.014	0.482±0.005	0.566±0.034	0.591±0.043	0.583±0.023	0.578±0.032	**0.593±0.020**

Table 4: Accuracy and Macro-F1 on Mhealth dataset. We bold the best score and underline the second best.

Datasets	Metrics	DeepConvLSTM [33]	XGBoost [7]	MA-CNN [36]	HHAR-net [14]	TST [52]	TARNet [10]	Rocket [12]	THAT [24]	SHARE
Mhealth	Accuracy	0.868±0.023	0.809±0.011	0.839±0.010	0.854±0.020	0.863±0.007	0.895±0.042	0.902±0.006	0.907±0.019	**0.975±0.014**
	Macro-F1	0.871±0.023	0.775±0.023	0.834±0.009	0.811±0.022	0.863±0.007	0.892±0.039	0.908±0.007	0.910±0.017	**0.974±0.013**

Table 5: Different model variants on Mhealth dataset. We bold the best score and underline the second best.

Datasets	Metrics	no token aug	no embed aug	no seq aug	SHARE
Mhealth	Accuracy	0.968±0.027	0.949±0.019	0.908±0.008	**0.975±0.014**
	Macro-F1	0.969±0.027	0.949±0.021	0.905±0.012	**0.974±0.013**

Table 6: Original/generated label names for Mhealth data.

Original Label Names	Generated Label Names
standing still	leg still
sitting and relaxing	buttocks still
lying down	back down
walking	leg walk
climbing stairs	leg up
waist bends forward	back forward
frontals elevation of arms	arm up
knees bending (crouching)	leg forward
cycling	leg cycle
jogging	leg jog
running	leg jog fast
jump front and back	leg jump

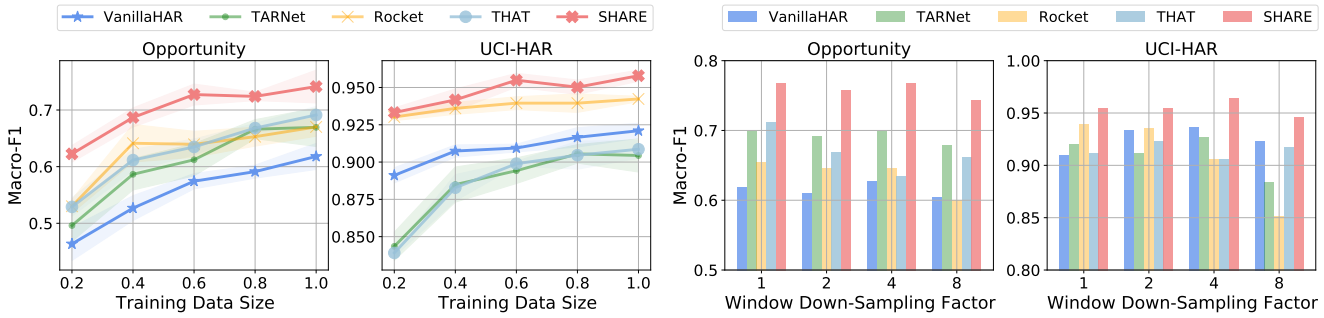
learning-based human activity recognition and time-series classification approaches, in terms of both accuracy and macro-F1 score. SHARE reduces the error rate (i.e., 1 - accuracy) on six datasets by approximately 20%, 30%, 34%, 9%, 9%, 11% compared with each dataset's best-performing baseline. Compared with the hierarchical baseline HHAR-net which models activities in a simple 2-layer hierarchical model, SHARE can model much more complex dependencies not necessarily in a hierarchical structure (e.g., "open door", "open drawer", "close drawer" with pairwise overlap, forming a graph rather than tree structure), without the cost of manual labeling from experts. TST and TARNet leverage unsupervised representation learning to boost classification performance. However, they do not explicitly take account of label structures to model relations across different activities. Other top-performing HAR or time-series classification methods, such as Rocket and THAT, propose better feature extractors to improve recognition performance, but they also neglect the label name structures. SHARE is capable of leveraging the inherent shared structures in label names, leading to the highest accuracy and macro-F1 score.

To assess the statistical significance of the performance differences between SHARE and the baselines, we applied the Wilcoxon-signed rank test with Holm's α (5%) following the procedures described in ShapeNet [17, 26]. The Wilcoxon-signed rank test indicates that the improvement of SHARE compared with all the baselines is statistically significant with p far below 0.05 (e.g., $p = 5e^{-4}$ for the best-performing baseline THAT).

5.4 Model Variants

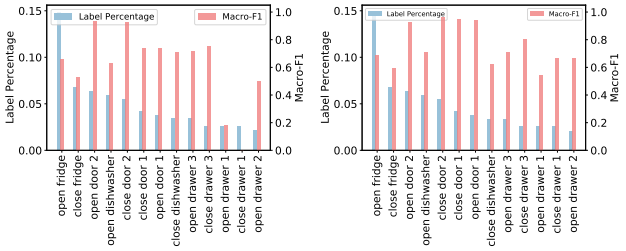
We also compare SHARE with some of its variants to examine the source of the performance gain. For all variants, we use the same encoder for feature extraction as SHARE.

- **VanillaHAR**: We use the same encoder as SHARE to extract features embedded in the data, and directly append a linear layer for classification without label name modeling.
- **VanillaHAR + ImageBind embeddings**: We also try directly incorporating ImageBind embeddings into VanillaHAR. This variant has two separate linear branches at the end. One branch is for classifying the labels, and the other branch predicts embeddings for the label names. During training, apart from the classification cross-entropy loss, we maximize the cosine similarity between the predicted embeddings and the pre-trained ImageBind embeddings. If the label names have multiple words, we use the average ImageBind embedding of each word as the embedding for the entire label name sequence.
- **multi-label classification**: We also try two separate classifiers subsequent to the encoder. The first classifier predicts the original labels, and the second operates as a multi-label classifier that estimates individual tokens within the label sequences. For example, to predict the class "walk forward", the second classifier labels "walk" and "forward" as positive and other tokens as negative.



(a) Reduced Training Sample (b) Reduced Window Size

Figure 6: Macro-F1 of SHARE, VanillaHAR and best-performing baselines with reduced training samples and window size.



(a) VanillaHAR (b) SHARE

Figure 7: Macro-F1 of example activities with shared label names for SHARE and VanillaHAR on Opportunity dataset with long-tail label distribution.

Classification of shared tokens helps learn dependencies across activities, and during testing, we only compare scores from the first classifier for original activity classes.

- **no aug**: We stay with the label structure decoding architecture but remove all three label augmentations.
- **no token aug**: We stay with the label structure decoding architecture but remove token-level augmentation during training.
- **no embed aug**: We randomly initialize the decoder word embedding layer instead of using ImageBind word embeddings.
- **no seq aug**: The Mhealth dataset [4] (publicly available at UCI Machine Learning Repository[7]) rarely has shared tokens in its original label names. We compare the performance of SHARE on its original non-overlapping label names and pre-trained model-augmented shared label names.

As shown in Table 3, we observe significant improvement from only applying a feature encoder to the proposed label structure architecture that decodes label names. Regressing label name embeddings by optimizing a cosine similarity loss with ImageBind embeddings only slightly improves the performance. This demonstrates that directly incorporating word embeddings does not explicitly take into account the shared label name structures and loses information when aggregating multiple words into a single label embedding. By contrast, SHARE generates label sequences which preserves the label structures and encourages knowledge sharing across activities. Compared with multi-label classification, our label structure decoding approach can preserve the word order (especially for multi-gram) and word correlation in label sequence.

[7]http://archive.ics.uci.edu/ml/datasets/mhealth+dataset

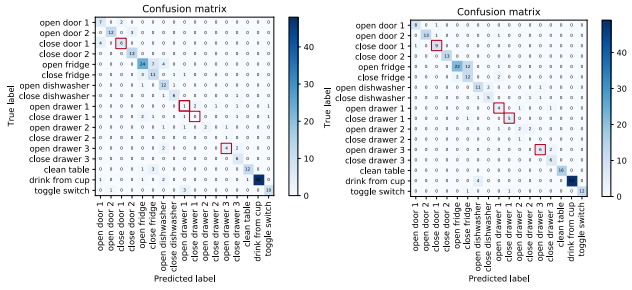
Moreover, the performances degrade after removing either token-level or embedding-level augmentation (or removing both), which validates their importance in capturing shared word semantics. For sequence-level augmentation, we summarize the original and generated label names from pre-trained model (GPT-4) in Table 6. We compare SHARE using generated label names against both baselines (Table 4) and our model variants (Table 5) on the Mhealth dataset. With the help of the automated label generation method, SHARE demonstrates state-of-the-art performance for HAR datasets without original shared label names. Moreover, we observe that sequence-level augmentation and embedding-level augmentation serve as complementary strategies that synergistically enhance performance.

5.5 Few-Shot Settings

We further evaluate SHARE under various few-shot settings.

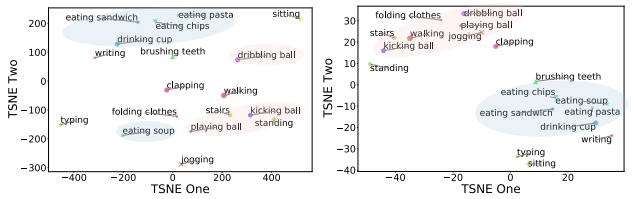
Reduced Training Samples. We randomly reduce the number of samples in the training set from two HAR datasets (Opportunity and UCI-HAR) to 20%, 40%, 60%, and 80%, and evaluate the macro-F1 on the same original test set. We conducted the experiments for 5 runs and report both average Macro-F1 as well as standard deviation. Figure 6a illustrates the performance trend of SHARE, VanillaHAR as well as the best-performing baselines when we vary the size of the training set. As we could observe from the figure, the macro-F1 generally increases as the number of available training samples increases. On top of that, the performance gap between SHARE and other methods becomes larger when there are fewer training data available, showing that decoding label names helps learn the common structures that are shared across different classes.

Label Imbalance. The above experiment reduces training samples for all the classes. Many HAR datasets also naturally have a long-tail distribution where some activities have fewer samples as being more difficult to collect. We also experiment under such label imbalance scenarios as shown in Figure 7. We compare SHARE and the vanilla classification model VanillaHAR by visualizing example activities with shared tokens. The activity names are sorted in decreasing order by the label percentage in the dataset. The performance grows significantly when adopting the label structure decoding architecture, as decoding label names helps transfer the shared word semantics to those classes with fewer available samples. For example, for the tail classes "open drawer 1", "close drawer 1", "open drawer 2", VanillaHAR shows a low F1 score (even zero for "close drawer 1"), while SHARE substantially improves the



(a) VanillaHAR (b) SHARE

Figure 8: Confusion matrix of VanillaHAR and SHARE on Opportunity dataset. SHARE better discriminates different activities, exemplified by classes with red squares.



(a) VanillaHAR (b) SHARE

Figure 9: T-SNE visualization on feature space. SHARE better preserves the semantics in the feature space.

performance on these classes, as SHARE is able to leverage label structures to learn from other classes.

Reduced Window Size. We also reduce the sampling frequency (window size) on both training and test sets by a factor of 2,4,8 and report the performance of SHARE, VanillaHAR as well as the best-performing baselines in Figure 6b. SHARE also stays robust with respect to down-sampling factors, as it encourages knowledge transfer via modeling label name structures. We observe that our proposed SHARE consistently outperforms VanillaHAR and baselines under different down-sampling factors.

5.6 Case Study

In this section, we further explore the benefits of modeling label structures through some case studies.

Confusion Matrix. We use Opportunity dataset as an example and show through the confusion matrix in Figure 8 that SHARE better discriminates activities compared with VanillaHAR, especially for activities with fewer samples. In Figure 8, values at the i-th row and j-th column represent the number of instances that have ground truth label i and are predicted as label j. "open drawer 1" instances mispredicted as "close drawer 1" are reduced from 2 to 0, and the correctly predicted instances increase from 1 to 4.

Feature Embedding. We apply t-SNE visualization to the feature space of VanillaHAR and SHARE on the WISDM dataset. We visualize the average feature of each activity, as illustrated in Figure 9. VanillaHAR loses the semantic information in the feature space. For example, "eating soup" is positioned at a large distance from other "eating"-related activities. By contrast, SHARE preserves the label structures in the feature space, indicating a more coherent and precise mapping of related activities.

Table 7: Model complexity analysis.

Model	# of Params	Model Size	Avg Running Time Per Batch
TST	1.195M	4.786MB	0.014s
TARNet	0.310M	2.465MB	0.016s
THAT	3.207M	12.828MB	0.018s
SHARE	**0.219M**	**0.878MB**	**0.003s**

5.7 Complexity Analysis

We compare the model complexity of SHARE and the best-performing deep models TST, TARNet and THAT on PAMAP2 data. Specifically, we compute the number of parameters, the model size (number of bytes required to store the parameters in the model), and the average running time for a batch of 16 samples (averaged over 10000 runs). We conducted the complexity analysis on a single NVIDIA RTX A6000 48G GPU. For TST, we only compare the complexity for the supervised fine-tuning phase. As shown in Table 7, SHARE has the smallest number of parameters, model size, and average running time, while outperforming more complex deep models.

6 CONCLUSION

We proposed a novel HAR approach, SHARE, that explicitly models the semantic structure of class labels and classifies the activities by decoding label sequence. SHARE enables knowledge sharing across different activity types via label name modeling and alleviates the challenges of annotated data shortage in HAR, compared with conventional methods that treat labels simply as integer IDs. We also design three label augmentation techniques, at token, embedding and sequence levels, to help the model better capture semantic structures across activities. We evaluated SHARE on seven HAR benchmark datasets, and the results demonstrate that our model outperforms state-of-the-art methods.

In the future, we plan to adapt our design to more complex backbone models, as well as image-based or video-based human activity recognition. We also plan to experiment on other types of datasets that also have shared label name structures , e.g., medical datasets with shared disease names. Also, in this work, we assumed that the shared label name structures very likely imply similarity in activity types. However, the assumption may not hold when we extend the problem scope to simultaneously handling multiple datasets where the same label names may correspond to slightly different data collection settings. We believe further investigation to lift such an assumption will offer meaningful insights.

7 ACKNOWLEDGEMENTS

Our work is supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Our work is also supported by Qualcomm Innovation Fellowship and is sponsored by NSF CAREER Award 2239440, NIH Bridge2AI Center Program under award 1U54HG012510-01, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

REFERENCES

[1] 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.

[2] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. 2009. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2009), 436–446.

[3] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

[4] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. 2014. mHealthDroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*. Springer, 91–98.

[5] Kaixuan Chen, Lina Yao, Dalin Zhang, Xiaojun Chang, Guodong Long, and Sen Wang. 2019. Distributionally robust semi-supervised learning for people-centric sensing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33.

[6] 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 Computing Surveys (CSUR)* 54, 4 (2021).

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[8] Belkacem Chikhaoui and Frank Gouineau. 2017. Towards automatic feature extraction for activity recognition from wearable sensors: a deep learning approach. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE.

[9] Ranak Roy Chowdhury, Jiacheng Li, Xiyuan Zhang, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. 2023. PrimeNet: Pre-Training for Irregular Multivariate Time Series. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[10] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. 2022. TARNet: Task-Aware Reconstruction for Time-Series Transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA*. 14–18.

[11] Zhicheng Cui, Wenlin Chen, and Yixin Chen. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016).

[12] Angus Dempster, François Petitjean, and Geoffrey I Webb. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* 34, 5 (2020), 1454–1495.

[13] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. 2021. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 248–257.

[14] Mehrdad Fazli, Kamran Kowsari, Erfaneh Gharavi, Laura Barnes, and Afsaneh Doryab. 2021. HHAR-net: hierarchical human activity recognition using neural networks. In *International Conference on Intelligent Human Computer Interaction*. Springer, 48–58.

[15] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. 2010. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing* 14, 7 (2010), 645–662.

[16] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016).

[17] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.

[18] HM Sajjad Hossain and Nirmalya Roy. 2019. Active deep learning for activity recognition with context aware annotator selection. In *KDD*. 1862–1870.

[19] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery* 34, 6 (2020), 1936–1962.

[20] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. 2019. SenseHAR: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*.

[21] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 2017. LSTM fully convolutional networks for time series classification. *IEEE access* 6 (2017), 1662–1669.

[22] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. 2019. Multivariate LSTM-FCNs for time series classification. *Neural Networks* 116 (2019), 237–245.

[23] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. 2015. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE international conference on computer vision*. 4247–4255.

[24] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. 2021. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 286–293.

[25] Chenglin Li, Di Niu, Bei Jiang, Xiao Zuo, and Jianming Yang. 2021. Meta-har: Federated representation learning for human activity recognition. In *Proceedings of the Web Conference 2021*. 912–922.

[26] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace LH Wong. 2021. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 8375–8383.

[27] Jason Lines, Sarah Taylor, and Anthony Bagnall. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data* 12, 5 (2018).

[28] Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. 2021. HARTH: A Human Activity Recognition Dataset for Machine Learning. *Sensors* 21, 23 (2021), 7853.

[29] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. 2019. AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition.. In *IJCAI*. 3109–3115.

[30] Yuchao Ma and Hassan Ghasemzadeh. 2019. LabelForest: Non-parametric semi-supervised learning for activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4520–4527.

[31] Moe Matsuki, Paula Lago, and Sozo Inoue. 2019. Characterizing word embeddings for zero-shot sensor-based human activity recognition. *Sensors* 19, 22 (2019), 5043.

[32] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023).

[33] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[34] Hangwei Qian, Sinno Pan, and Chunyan Miao. 2018. Sensor-based activity recognition via learning from distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[36] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2018. Multimodal deep learning for activity and context recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–27.

[37] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*. IEEE, 108–109.

[38] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.

[39] Patrick Schäfer and Ulf Leser. 2017. Multivariate time series classification with WEASEL+ MUSE. *arXiv preprint arXiv:1711.11343* (2017).

[40] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I Webb. 2020. TS-CHIEF: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery* 34, 3 (2020), 742–775.

[41] Mohammad Shokoohi-Yekta, Jun Wang, and Eamonn Keogh. 2015. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM international conference on data mining*. SIAM, 289–297.

[42] Catherine Tong, Jinchen Ge, and Nicholas D Lane. 2021. Zero-Shot Learning for IMU-Based Activity Recognition Using Video Embeddings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–23.

[43] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. 2019. Informed Machine Learning–A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *arXiv preprint arXiv:1903.12394* (2019).

[44] Wei Wang, Chunyan Miao, and Shuji Hao. 2017. Zero-shot human activity recognition via nonlinear compatibility based method. In *Proceedings of the International Conference on Web Intelligence*. 322–330.

[45] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*. IEEE, 1578–1585.

[46] Gary M Weiss, Kenichi Yoneda, and Thaier Hayajneh. 2019. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access* 7 (2019), 133190–133202.

[47] Waskitho Wibisono, Dedy Nur Arifin, Baskoro Adi Pratomo, Tohari Ahmad, and Royyana M Ijtihadie. 2013. Falls detection and notification system using tri-axial accelerometer and gyroscope sensors of a smartphone. In *2013 Conference on Technologies and Applications of Artificial Intelligence*. IEEE, 382–385.

[48] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[49] Tong Wu, Yiqiang Chen, Yang Gu, Jiwei Wang, Siyu Zhang, and Zhanghu Zhechen. 2020. Multi-layer cross loss model for zero-shot human activity recognition. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 210–221.

[50] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. 351–360.

[51] Zhi Zeng and Qiang Ji. 2010. Knowledge based activity recognition with dynamic bayesian network. In *European conference on computer vision*. Springer, 532–546.

[52] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2114–2124.

[53] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. 2022. Towards Similarity-Aware Time-Series Classification. *arXiv preprint arXiv:2201.01413* (2022).

[54] Jiayun Zhang, Xiyuan Zhang, Xinyang Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. 2023. Navigating Alignment for Non-identical Client Class Sets: A Label Name-Anchored Federated Learning Framework. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

[55] Mi Zhang and Alexander A Sawchuk. 2012. USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 1036–1043.

[56] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. 2021. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *arXiv preprint arXiv:2111.00418* (2021).

[57] Xiyuan Zhang, Ranak Roy Chowdhury, Jingbo Shang, Rajesh Gupta, and Dezhi Hong. 2022. ESC-GAN: Extending spatial coverage of physical sensors. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1347–1356.

[58] Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. 2020. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6845–6852.

[59] Fengtao Zhou, Sheng Huang, and Yun Xing. 2021. Deep semantic dictionary learning for multi-label image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3572–3580.