Contrasting Global and Local Representations for Human Activity Recognition using Graph Neural Networks

Andrés Tello
Bernoulli Institute, University of Groningen
Groningen, The Netherlands
andres.tello@rug.nl

Victoria Degeler
Informatics Institute, University of Amsterdam
Amsterdam, The Netherlands
v.o.degeler@uva.nl

Abstract

Human Activity Recognition has achieved notable improvements with the emergence of Deep Learning models for automated feature extraction. Those models allow to extract complex translational-invariant features and to exploit the temporal dependencies from sensors' time series data. This work posits additional dependencies between sensors beyond the time dimension, such as physical proximity, which are equally important for the characterization of human activities. We leverage such spatial dependencies by modeling them as a graph. Using Graph Neural Networks (GNNs), we learn global and local representations of the intra- and inter-sensor dependencies. We empirically show that by maximizing the mutual information between the local and global representations, the performance of the recognition models can be significantly improved. Our results show a clear improvement over previous works based on CNNs, LSTMs, Attention-based and other more complex GNNs-based architectures. Our source code is available at: https://github.com/atello/GNNs4HAR

CCS Concepts

• **Computing methodologies** → **Semi-supervised learning settings**; **Learning latent representations**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

Keywords

Graph Neural Networks, Graph Classification, Graph Contrastive Learning, Human Activity Recognition

ACM Reference Format:
Andrés Tello and Victoria Degeler. 2025. Contrasting Global and Local Representations for Human Activity Recognition using Graph Neural Networks. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3672608.3707743

1 Introduction

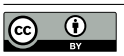
Human Activity Recognition (HAR) is a field that has attracted the attention of Academia and Industry for several years. In the early years of HAR, the problem of recognizing human activities was addressed using classical machine learning approaches, e.g., Decision Trees, Support Vector Machines, Naïve Bayes and k-Nearest Neighbors [1, 2, 28]. Later on, Deep Learning approaches based on Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) have been extensively applied in the field of HAR, showing significant results [9, 14, 26, 32, 39]. Those approaches mainly exploit the relationships across the data in the time dimension. On the one hand, CNNs are used for feature extraction to find patterns which are invariant to translations across different segments [14]. On the other hand, LSTMs, a type of RNN, only exploit temporal dependencies [26].

The current study, in addition to the temporal patterns, also leverages the spatial dependencies in the data and introduces them as relational inductive biases represented as a graph structure. The assumption is that signals collected from smartphones and wearables, while people perform different activities, have additional dependencies that span beyond the temporal dimension. These spatial dependencies between intra- and inter-sensor's data channels can be represented as a graph. GNNs have been successfully applied to learn complex relationships in graph-structured data [4, 11–13]. Then, such relationships can be learned by a GNN-based model and subsequently used to characterize human activities.

Although previous work introduced the idea of applying GNN-based models in the HAR domain [18, 33, 37], these do not reflect the real performance of the models due to a methodological issue within the HAR pipeline that causes biased results. Such biases are introduced by the evaluation strategy that follows a sliding-windows data segmentation approach [15, 30]. Therefore, the real value that GNN-based models can bring to the field was not clearly established from these works.

In this paper, we formulate HAR as a multi-class graph classification problem. We apply two different graph construction methods from raw sensor signals, which leverage *global* and *local* dependencies in the time series data. While global dependencies are captured by correlating the sensor's channels from the entire sequence, local dependencies capture the correlations within smaller time frames. Thus, global and local graphs represent different views of the same human activities. Then, a contrastive learning approach, to maximize mutual information between the two views, shows a significant performance gain for HAR models.

We evaluate our proposed approach on four benchmark datasets, UCIHAR [1], MHEALTH [2], PAMAP2 [28], and REALDISP [3]. These datasets comprise accelerometer, gyroscope and magnetometer data from Inertial Measurement Unit (IMU) sensors. Our GNNs-based models produce accuracies over 90% on UCIHAR, MHEALTH and REALDISP datasets, and ≈ 87% accuracy on PAMAP2. These results surpass previous approaches based on CNNs, LSTMs, Attention-based and other more complex GNNs-based architectures. It shows that GNNs has potential in the HAR domain, allowing to discover different relationships in the data beyond the temporal dimension.



SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707743

The main contributions of our work are: (i) A new set of state-of-the-art GNNs-based models for HAR with significant reduced complexity compared to existing approaches, (ii) Global and local graph representations of the inter- and intra-sensor dependencies, and (iii) A contrastive learning approach that leverages the global/local dependencies to enhance the performance of HAR models, showing a clear improvement over other state of the art methods.

The remainder of this document is as follows. Section 2 shows the related work to this field and discusses the main differences of our work with the existing GNNs-based approaches for HAR. Section 3 describes and formalizes our approach. Section 4 describes the experiments performed on Human Activity Recognition using GNNs. Section 5 presents and discusses the results. Finally, Section 6 presents the conclusions.

2 Related Work

Most of the existing work on HAR using GNNs is based on skeleton graphs extracted from video sequences [8, 21, 36]. During the last years, the community started to explore the capabilities of GNNs for HAR using IMU sensors from smartphones or other types of wearable devices [18, 24, 33, 37].

Huang et al. [18] propose a shallow CNN that performs cross-channel communication for HAR. The cross-channel communication is achieved by message passing within a GNN. They propose a 3-layer CNN followed by a fully connected layer for final classification. However, they encode the output of each CNN layer via graph attention network over a fully connected graph where each channel of the CNN is considered a node. Then, the encoded signals are send back to the next CNN layer. They evaluated their proposed approach by comparing a 3-layers CNN, 6-layers CNN and a 3-layers CNN + GAT. The results show that cross-channel communication via GNN message passing outperforms a deeper CNN architecture. Since our architecture is based only on a 3-layer GNN, the number of parameters of our model is reduced by half with respect to this work.

Mohamed et al. [24] propose HAR-GCNN, a deep graph CNN to classify human activities. Their approach is based on the assumption that people perform certain activities in a chronological order. Hence, they exploit the correlation between chronologically adjacent activities to predict unlabeled or future activities. They create a fully connected graph by taking a number of consecutive samples in a time window t, where each sample represents a node in the graph. Then, they randomly mask a number of activity labels and add noise to the features of some randomly selected nodes. The model is trained to predict the missing labels. Taking n consecutive activities and randomly masking some of the labels is not realistic because the nodes representing future activities may be used to predict the past ones. In a realistic implementation, only the most recently executed activities can be used to predict the next one (or at most n in the near future). Hence, masking can not be made at random. In addition, the authors assume that activities are usually executed by people in a certain particular order. Such assumption can lead to models that learn the sequence in which activities were performed rather than real patterns that characterize each activity [7]. This effect is more pronounced in datasets with strict data collection protocols [7], but can be less noticeable in a more natural setting with a loose order of activities. From the results reported in this work, it is clear that the model is learning the order on which the activities were performed. The results on PAMAP2 dataset, which has a strict data collection protocol, are almost perfect. On the contrary, the performance drops ≈10% points on Extra-Sensory dataset, in which subjects were not instructed on how to perform the activities. Our work does not make any assumption on the order on how the activities were performed. Thus, the graph construction does not depend on the sequence of data samples but on the correlations between sensors' data signals.

Yan et al. [37] propose a model based on 4 blocks composed of 4 Chebyshev [10] layers, followed by a normalization layer and a Leaky ReLU activation function. At the end, two fully connected layers are used as a classifier. Their model architecture is composed of 18 layers with 5.29M trainable parameters in total. They also used the MHEALTH and PAMAP2 datasets. The raw input data signals are transformed into a graph representation based on Pearson correlation coefficient matrix. A correlation above 0.2, as a threshold, implies an edge between those two signals. The authors reported accuracy above 98% for PAMAP2 and MHEALTH datasets. The work of Wang et al. [33] is build upon [37]. It also computes the adjacency matrix for the graph representations based on the Pearson correlation coefficient. What makes this work interesting is the combination of message passing GNNs with multi-head attention to learn channel-wise correlations.

Unfortunately, some results presented in the aforementioned works, [33, 37], are misleading due to accuracy overestimation caused by a biased evaluation protocol. In those studies, the data segmentation is performed using the well established sliding-windows approach. The problem arises when the training/validation/test sets are determined at random. The sliding-windows approach introduces a high correlation between two consecutive windows and therefore the way in which training and test samples are chosen affects the performance of the models, as discussed in detail in [15, 30]. We show that, with a proper evaluation protocol, the performance reported in [37] drops 14 percentage points on average. This highlights the need for a thorough exploration of GNN-based models for HAR.

The main difference of our work with these approaches is that we add a local graph representation computed on per-window Pearson correlation coefficients. Then we contrast the global and local representations to maximize the mutual information between them. Since our model is shallow, the complexity is reduced to 0.43M parameters, achieving a significantly higher performance.

3 Graph Classification for Human Activity Recognition

In this section we describe and conceptualize our approach. First we describe how we model human activities as graphs. Then we describe how we formulate HAR as a graph classification problem.

3.1 Human Activities as Graphs

Given the raw data signals from the IMU sensors in the form of a matrix $X \in \mathbb{R}^{n \times m}$, where n is the number of observations and m is the number of channels in the raw signals, an activity graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, l)$. The set of nodes or vertices is given

by $\mathcal{V}$, where each vertex represents a channel of the signal. The edge set is given by $\mathcal{E} = \{(i,j) \in \mathcal{V} \times \mathcal{V} : w_{ij} \neq 0\}$, where w_{ij} is the weight of the edge connecting node i to node j, i.e., the strength of the connection between two signal channels. The label, $l \in \mathcal{L}$, denotes the activity associated with a graph $\mathcal{G}$, where $\mathcal{L}$ is the set of all activity labels in the dataset.

The nodes $\mathcal{V}$, given by the different channels of the signals, are known beforehand. The edge set $\mathcal{E}$, and its associated weights w_{ij}, have to be estimated from data observations. Initially we assume a fully connected graph, and the number of edges $|\mathcal{E}| = m(m-1)$. Then, the edge set is refined by choosing a subset of edges from $\mathcal{E}$ based on data statistics e.g., covariance/kernel matrices.

In our work, we used the correlation matrix to build the graph topology and estimate the edge weights from the raw data signals. Similar to [37] and [33], we rely on the Pearson's correlation coefficients matrix. This method has been widely used to discover the topology and estimate functional connectivity between the regions of the brain, and it is recognized to be one of the most used methods in that field [17, 27, 34]. The Pearson's correlation coefficient is defined as the covariance of two random variables, in our case $v_i, v_j \in \mathcal{V}$, normalized by the product of their standard deviations.

$$\rho_{(v_i,v_j)} = \frac{cov(v_i, v_j)}{\sigma v_i \, \sigma v_j} \tag{1}$$

Using equation (1) we obtain a correlation matrix $X_{corr} \in \mathbb{R}^{m \times m}$, where a zero value implies no (linear) correlation between two random variables, i.e., there is an edge connecting two nodes *if and only if* the correlation coefficient is different than zero. The correlation coefficients in X_{corr} gives us the strength of the correlation; hence, they can be used as the weights of the edges connecting a pair of nodes in our activity graphs. However, instead of using the complete correlation matrix for computing the edges and their weights, we used a sparse version by setting a threshold τ. The threshold allows to filter out noisy connections in our activity graphs. According to [27], when using Pearson's correlation to define the graph connectivity, the threshold can act as a L1-regularization term included in others approaches, e.g., Graphical Lasso. Taking into account that the correlation coefficients can be negative, the edge weight can be defined as follows:

$$w_{ij} = \begin{cases} |\rho_{(v_i,v_j)}|, & \text{if } |\rho_{(v_i,v_j)}| > \tau. \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The correlation coefficient values are considered weak for values $|\rho_{(v_i,v_j)}| < 0.2$, moderate for values of $0.2 \leq |\rho_{(v_i,v_j)}| \leq 0.3$ and strong for values of $|\rho_{(v_i,v_j)}| > 0.3$ [16, 29]. Therefore, the threshold τ was set to 0.2 to filter out weak connections and include moderate and strong categories in terms of correlation strength.

Applying the approach described above, we define two different types of activity graphs: $\mathcal{G}_{global}$ and $\mathcal{G}_{local}$. The first type of graphs, $\mathcal{G}_{global}$, are based on a single correlation matrix calculated from the entire training set. Then this matrix is used to define the edges and weights for all activity graphs. The difference between the activity graphs constructed using this approach is in the feature values of the nodes. The features for each node are given by the sensor readings within a temporal window t. For the second type of graphs, $\mathcal{G}_{local}$, a different graph is created for every time window t. Hence, a correlation matrix, $X^{(t)}_{corr}$, is calculated from the signal of the channels for each window and such matrix is used to define the edges and the weights of each graph. The motivation to create a different graph for each time window is that every activity is expected to have an underlying topology that describes the structural relationships between the signals. Therefore, since every time window corresponds to a specific activity, the model can learn the patterns hidden per activity class. This procedure is repeated for all the windows in the training, validation and test sets.

3.2 Classification of Human Activity Graphs

We formulated HAR as a supervised multi-class graph classification problem. In the activity graph classification setting, given a set of graphs $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_N\} \subseteq \mathbb{G}$ with its corresponding activity labels $\{l_1, l_2, ..., l_N\} \subseteq \mathbb{L}$, a GNN-based classifier should learn a vector representation of a graph $h_{\mathcal{G}}$ that allows to predict its associated label, $l_{\mathcal{G}} = g(h_{\mathcal{G}})$ [35].

In general terms, GNNs learn vector representations of a node, h_{v_i}, by iteratively aggregating the vector representations of its neighbors, $h_{v_j} : v_j \in \mathcal{N}(v_i)$, and then combining the aggregated vectors with its own representation at iteration $t-1$. Using the notation presented in [35], this iterative learning process can be denoted as:

$$a^{(t)}_{v_i} = AGGR^{(t)}(\{h^{(t-1)}_{v_j} : v_j \in \mathcal{N}(v_i)\}) \tag{3}$$

$$h^{(t)}_{v_i} = COMBINE^{(t)}(h^{(t-1)}_{v_i}, a^{(t)}_{v_i}) \tag{4}$$

Both *AGGR* (equation 3) and *COMBINE* (equation 4), may be arbitrary differentiable, permutation-invariant functions [25]. In a node classification problem, the final representation of a node ($h^{(T)}_{v_i}$) would suffice. For graph classification, we need a global function to aggregate the final representation of all nodes $v_i \in \mathcal{V}$. Once again, any permutation invariant function can be used, e.g., *sum* or *max* or other more advanced readout implementations like DiffPool [38] or SortPool [40]. Then, the vector representation of an entire graph can be defined as:

$$h_{\mathcal{G}} = READOUT(\{h^{(T)}_{v_i} : v_i \in \mathcal{G}\}) \tag{5}$$

The final vector representation of our activity graphs, $h_{\mathcal{G}}$ can be passed to a final classifier, e.g., a Multi Layer Perceptron (MLP), followed by a softmax function to predict the label associated to a graph, $l_{\mathcal{G}} = g(h_{\mathcal{G}})$.

3.3 Contrasting global and local activity graphs representations

We propose a contrastive learning algorithm based on the two different views of our activity graphs, $\mathcal{G}_{global}$ and $\mathcal{G}_{local}$, described in section 3.1. The idea is to maximize the mutual information between pairwise representations of the same activity. Thus, two separate GNN-based encoders compute the global and local vector representations (embeddings) of an activity graph by means of equations 3, 4, and 5. Then, the tuple $(h_{\mathcal{G}_{global}}, h_{\mathcal{G}^{+}_{local}})$, represent the positive pairs. Using the same local encoder to compute the embeddings of a corrupted version of $\mathcal{G}_{local}$, the negative pairs for contrasting are $(h_{\mathcal{G}_{global}}, h_{\mathcal{G}^{-}_{local}})$. The corruption of $\mathcal{G}_{local}$ is performed by a

random permutation of the nodes, a random permutation of the edges, or both. Finally, both, the global and local embeddings of the activity graphs are concatenated and passed through two fully connected layers for classification. Figure 1 shows and overview of the proposed contrastive learning approach for HAR.

4 Experiments

First, using the $\mathcal{G}_{global}$ and $\mathcal{G}_{local}$ graph representations independently, we evaluated the ability of GNNs to classify the human activities. Then, combining both representations, we followed a contrastive learning approach to train a model that maximizes the mutual information between activity graphs that represent the same activities, or minimizes it otherwise.

4.1 Datasets

The datasets used in this study are UCIHAR [1], MHEALTH [2], PAMAP2 [28], and REALDISP [3]. They comprise accelerometer, gyroscope and magnetometer data collected from smartphones and wearable devices while people perform different activities.

UCIHAR. The data were collected from a smartphone placed on the waist of 30 volunteers performing the activities: *Walking, Walking Up, Walking Down, Sitting, Standing* and *Laying*. These data were collected at a constant sampling rate of 50Hz. Subjects 2, 6, 12, 19, and 26 were used for validation. Data from subjects 5, 8, 10, 14, 20, and 21 were used for testing and rest for training.

MHEALTH. This dataset contains data of 10 volunteers performing the activities: *Standing, Sitting, Lying down, Walking, Climbing stairs, Waist bends forward, Arms up, Knees Bending, Cycling, Jogging, Running* and *Jumping*. The data was collected with wearables placed at subjects' chest, right wrist and left ankle. The sampling rate was 50Hz. Two 2-lead ECG measurements on the chest were not used in the experiments. Data from subjects 6 and 10 were used for validation, subjects 2 and 9 for testing and the rest for training.

PAMAP2. The data were collected from 9 subjects using IMUs attached to the wrist, chest and ankle while performing everyday, household and sport activities. The sampling rate was 100Hz. This dataset includes eighteen different activities. However, in our experiments we only used the main twelve activities: *Lying down, Sitting, Standing, Walking, Running, Cycling, Nordic Walk, Walking Upstairs, Walking Downstairs, Vacuum Cleaning, Ironing* and *Rope Jumping*. Data from subjects 101 and 107 were used for validation, subjects 103 and 105 for testing and the remaining for training.

REALDISP. This dataset contains data collected from 17 subjects while performing 33 physical activities. In this study, we only used the activities *Walking, Jogging, Running, Jumping, Jump rope, Waist bends forward, Arms up, Knees to breast, Knees Bending,* and *Cycling* to be consistent with the type of activities in the previous datasets. The recordings were sampled at 50 Hz. Data from subjects 4, 6, 10, and 11 were used for validation; subjects 1, 7, 8, 9, 12, and 14 for testing and the remaining for training.

4.2 Data Preprocessing

We followed the conventional approach of splitting the data into training, validation and test sets as described in the previous section. The splits were performed following a stratified by subject approach, where subjects for each subset were chosen at random. Using this scheme the data is split into folds with non-overlapping subjects, where percentage of samples for each class is preserved. Splitting the data by subject ensures the independence between training and validation data [15, 30], which was not observed in previous works, such as ResGCNN [37]. Then, z-score normalization was applied to our training, validation and test sets.

Next, the data was segmented following the sliding window approach. Since the data was split by subject, the independence between sets is guaranteed, and the sliding windows approach does not introduce bias at the evaluation time. The sliding windows for UCIHAR and PAMAP2 datasets were created following the data segmentation protocol described in their original publications verbatim. In the first case, the data was segmented in windows of 2.56 seconds (i.e., 128 samples) with 50% overlap. For the PAMAP2 dataset, 5.12 seconds (i.e., 512 samples) with 1 second shift (i.e., 100 samples). In the case of the MHEALTH and REALDISP datasets, we followed the same protocol as for UCIHAR because the data were collected at the same sampling rate for both datasets.

4.3 Models Configuration

All our experiments are based on a 3-GraphConvLayer GNN [25], followed by a ReLU activation and dropout after each of the first two convolutions. Then, 2-Fully-Connected layers serve as the final classifier. We trained the model for 500 epochs and used early stopping

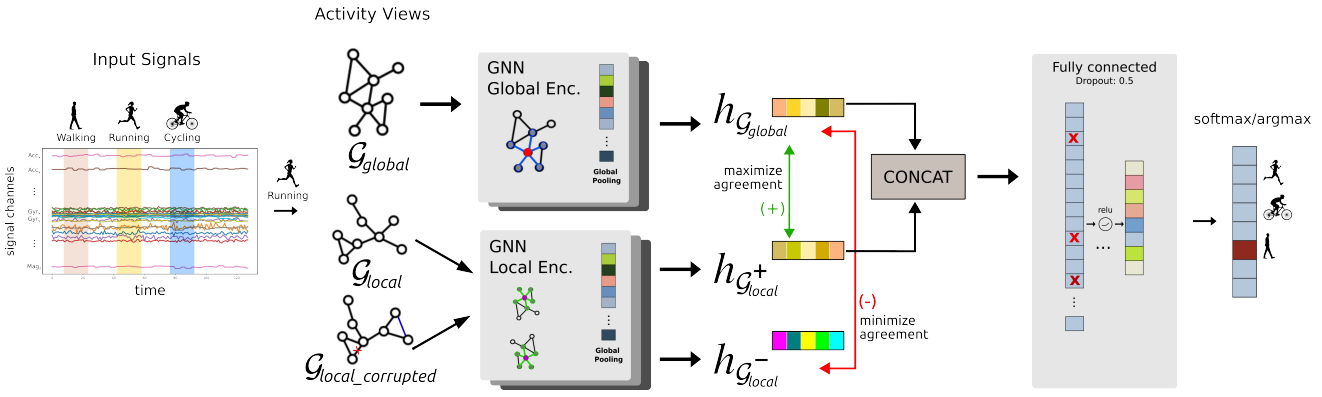


Figure 1: Contrastive global and local activity graph representations for HAR.

if the validation loss did not improve for 100 consecutive epochs. The models were optimized on the hyperparameters using the Tree-structured Parzen Estimator algorithm [5], implemented using the hyperopt python library [6]. The learning rate and weight decay were sampled from a *log uniform* distribution from the given lower and upper boundaries, and quantized to the specified increment value, i.e., the sampled value will be rounded to the nearest multiple of increment *inc*. The values in the hyperparameters search space were chosen in a way that they cover the values found in seminar GNNs-related papers [10, 20, 31]. The values in the search space for hyperparameters optimization are shown in Table 1.

Table 1: Hyperparameters optimization search space.

*learning rate	*weight decay	*layer dropout	*classifier dropout	hidden channels	batch norm	AGGR	READOUT
1e-4,	1e-5,	0.0,	0.0,	32,	True	add	add
1e-2	1e-3	0.6	0.6	64,	False	mean	mean
inc: 1e-5	inc: 1e-6	inc: 0.1	inc: 0.1	128		max	max

* min and max values of a range with increment *inc*

4.4 Loss functions

The baseline models, based on the $\mathcal{G}_{global}$ and $\mathcal{G}_{local}$ representations, were trained using the Adam optimizer [19] to minimize the Categorical Cross Entropy Loss, which is used for multi-class classification problems like the one presented in this work.

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p_{i,c}) \tag{6}$$

where N is the number of samples in the dataset or the batch size, C is the number of classes in the classification problem, $y_{i,c}$ is the ground truth label for sample i, and $p_{i,c}$ is the predicted probability for sample i for class c.

In the case of the contrastive learning approach, the models were trained on a composite loss function that combines the *classification loss* and the *contrastive loss*. Categorical Cross Entropy Loss (eq. 6) was used for the supervised classification, and InfoNCE loss (eq. 7) was used for the contrastive part.

$$\mathcal{L}_{\text{infoNCE}} = -\log\frac{\exp(\text{sim}(z_i,c_i))}{\exp(\text{sim}(z_i,c_i))+\sum_{j=1}^{N}\exp(\text{sim}(z_i,c_j))} \tag{7}$$

where N is the batch size, z_i is the representation of a global positive sample $h_{\mathcal{G}_{global}}$, c_i is the representation of the corresponding contrastive local positive sample $h_{\mathcal{G}_{local^+}}$, c_j is a corrupted version of c_i, representing the contrastive local negative sample $h_{\mathcal{G}_{local^-}}$, and $\text{sim}(z_i, c)$ is a similarity function (e.g., cosine similarity) between global (z_i) and local representations (c). Hence, the loss function for contrastive learning is given by:

$$\mathcal{L} = \mathcal{L}_{\text{infoNCE}} + \mathcal{L}_{\text{CE}} \tag{8}$$

4.5 Performance evaluation

After hyperparameter optimization, the models were updated with the entire training, and validation sets. The learning rate of the best trained model was reduced by a factor of 0.1 to avoid the model learning totally different weights. The model update is stopped when the validation loss does not improve for 10 epochs. The final model is evaluated on the accuracy and macro f1-score obtained on the holdout test set.

5 Results and Discussion

This section shows the results and discusses the findings obtained from the experiments executed in this work.

5.1 Different graph constructions.

The first set of experiments was performed using the global and local correlation matrices described in Section 3.1. The global representation captures the correlation between sensor data channels along the entire training data. On the contrary, the local model captures the correlations within a time window frame. Comparing just these two models (see Table 2), the results are consistent across all datasets in favor of the global model. However, the difference

Table 2: Classification accuracy and macro f1-score of the models on all datasets. Best in bold, second best underlined.

	Params	UCIHAR		MHEALTH		PAMAP2		REALDISP	
	(millions) ↓	acc↑	f1↑	acc↑	f1↑	acc↑	f1↑	acc↑	f1↑
CNN [32]	<u>0.09</u>	86.06	84.38	86.00	82.21	74.55	73.63	90.08	89.05
2-LSTM [23]	**0.08**	84.74	83.38	80.36	77.95	75.92	73.53	88.04	83.63
4-CNN-LSTM [23]	1.49	88.14	86.71	81.3	80.02	60.57	55.95	71.77	73.31
DeepConvLSTM [26]	2.92	89.42	87.82	83.74	79.75	80.53	77.92	88.26	89.34
Self-Attention [22]	0.43	85.00	85.00	79.00	78.00	83.00	81.00	72.00	68.00
ResGCNN [37]	5.29	83.13	83.33	86.00	84.03	82.18	81.97	77.75	74.35
Global (ours)	0.22	89.10	89.20	87.41	87.46	**86.86**	**86.36**	**93.65**	**91.52**
Local (ours)	<u>0.09</u>	88.42	88.80	86.56	86.61	81.29	83.12	84.03	82.59
contrastive$_{all}$ (ours)	0.43	<u>90.81</u>	<u>91.07</u>	89.66	89.82	80.44	82.28	90.37	89.91
contrastive$_{edges}$ (ours)	0.43	89.93	90.28	<u>90.32</u>	<u>90.44</u>	<u>83.71</u>	<u>84.64</u>	<u>92.12</u>	<u>91.02</u>
contrastive$_{nodes}$ (ours)	0.43	**91.43**	**91.82**	**90.60**	**91.00**	82.91	84.3	88.26	86.13

between the global and local models for UCIHAR and MHEALTH dataset is small, contrary to the difference found for PAMAP2 and REALDISP datasets. The PAMAP2 dataset was the only one containing missing data. The results reported for the Local model on the PAMAP2 dataset come from the data that were cleaned using interpolation to fill missing values. Hence, the data windows having mostly interpolated datapoints in the local model do not allow to properly define the topology and the strength of the connections, as opposed to the global model. Looking at global or local correlations independently, the global view allows GNNs to learn a better representation of the sensor data, even under the presence of noise like in the PAMAP2 dataset.

These results indicate that Pearson correlation is a viable option to extract the hidden topology between signals from wearable sensors. Figure 2 shows the confusion matrices of accuracy obtained with the Global model on all datasets. The model struggles to distinguish between standing and sitting activities in UCIHAR, MHEALTH, and PAMAP2. The model also confuses running and jogging activities in MHEALTH and REALDISP. This shows that there are common patterns, shared across all datasets, associated with a particular activity, and that GNN-based models are able to learn those patterns. Interestingly, the model shows consistency across datasets even for those activities that it fails to classify.

5.2 Contrastive learning

The results of contrastive learning experiments also show the consistency of the GNN-based models. A single perturbation, either to the nodes or edges, produced the best results in all cases. Compared to the local representation alone, combining it with the global one allows the models to learn stronger vector representations that boost the performance of the classifiers. Our contrastive learning approach is suitable for HAR as it allows the model to learn the shared and also the uncommon patterns between similar activities. The walking up and walking down activities involve the same body parts; and thus, they share common patterns. We argue that while the global model learns those shared patterns, the local model refines the learning process with the patterns that are intrinsic to each time window frame and related to the way the activity is performed. This can be observed in the confusion matrix calculated with the predictions of the *Global* model in UCIHAR. The *Global* model confuses the walking up and walking down activities (Figure 2a), which we attribute to the fact that these activities have similar global patterns. On the contrary, the contrastive learning approach on UCIHAR improves significantly on these two activities (Figure 3a). Similar behavior occurs in MHEALTH. The *Global* model confuses the standing and sitting, waist bend forward and knees bending, and jogging and running activities (Figure 2b). In this case, the contrastive learning model improves on the standing and sitting activities, and removes the confusion between waist bend forward and knees bending activities almost completely (Figure 3b).

The contrastive learning approach shows significant improvement when the global and local representation models have similar performance. This may be considered as a prerequisite for applying the contrastive learning approach. This is the case for UCIHAR and MHEALTH datasets where the contrastive model achieves 2.65 and 3.54 percentage points improvement, respectively. On the contrary, for PAMAP2 and REALDISP datasets, where the performance of global and local models was not on par, the contrastive learning approach did not contribute to performance gains.

The contrastive approach shows a promising direction for further exploration. In this approach, we contrasted different views of the graphs representing the same activities. Incorporating different graph augmentation techniques for creating a richer set of views to contrast is considered part of the future work.

5.3 Comparison with other Deep Learning models

We compared our approach with other Deep Learning approaches for HAR reported in literature. Namely, CNN [32], 2-LSTM [23], 4-CNN-LSTM [23], DeepConvLSTM [26], Self-Attention [22], and ResGCNN [37]. The results presented in Table 2 show that our *global* representation allows our model to outperform all other models in all datasets in terms of f1-score. In terms of accuracy, our *global* model surpasses all of the models on MHEALTH, PAMAP2, and REALDISP datasets by a large margin. On UCIHAR dataset, only DeepConvLSTM is on par with our model, while the others show worse results. Moreover, leveraging the mutual information between the global and local representations of the human activities, our contrastive learning approaches improves the performance even further, surpassing all other models on all of the tested datasets.

In terms of model complexity, our smaller model, *local*, has 0.09M parameters being on par with the CNN and 2-LSTM models. However, our model exhibits a clear performance gain in UCIHAR, MHEALTH and PAMAP datasets. Likewise, our *global* model, with 0.22M parameters, is smaller than all other counterparts, except CNN and 2-LSTM, but outperforms all of them in accuracy and f1-scores in all datasets.

Our results confirm the performance drop of ResGCNN [37] when model evaluation is correctly implemented, fixing the issue of the performance overestimation caused by random data splits leaking training data into the test set [15, 30]. We evaluated ResGCNN ensuring the independence between training, validation, and test sets by partitioning the data by subject as described in Section 4.2. The best result obtained with that model was a 86% accuracy for the MHEALTH dataset, which is significantly lower than the ≈98% reported in [37]. Our contrastive model shows a clear improvement with an accuracy of 90.60%. It is important to point out that the difference in complexity between our models and ResGCNN is very large. Our more complex model is the one used for Contrastive Learning with 0.43M parameters, while ResGCNN has 5.29M. Our model is ≈ 91% smaller but it achieves a much higher performance. This shows that very deep models, at least for these datasets, are not necessary. The performance decrease of ResGCNN shows that, even using residual connections, the oversmoothing problem, common in GNNs, still affects the learning capabilities of the model.

6 Conclusions

In this work, we leveraged the spatial dependencies between IMU sensors' data channels by modeling them as a graph. Using two GNNs-based encoders, we learned global and local representations of the intra- and inter-sensor dependencies, and exploited those representations by maximizing the mutual information between them

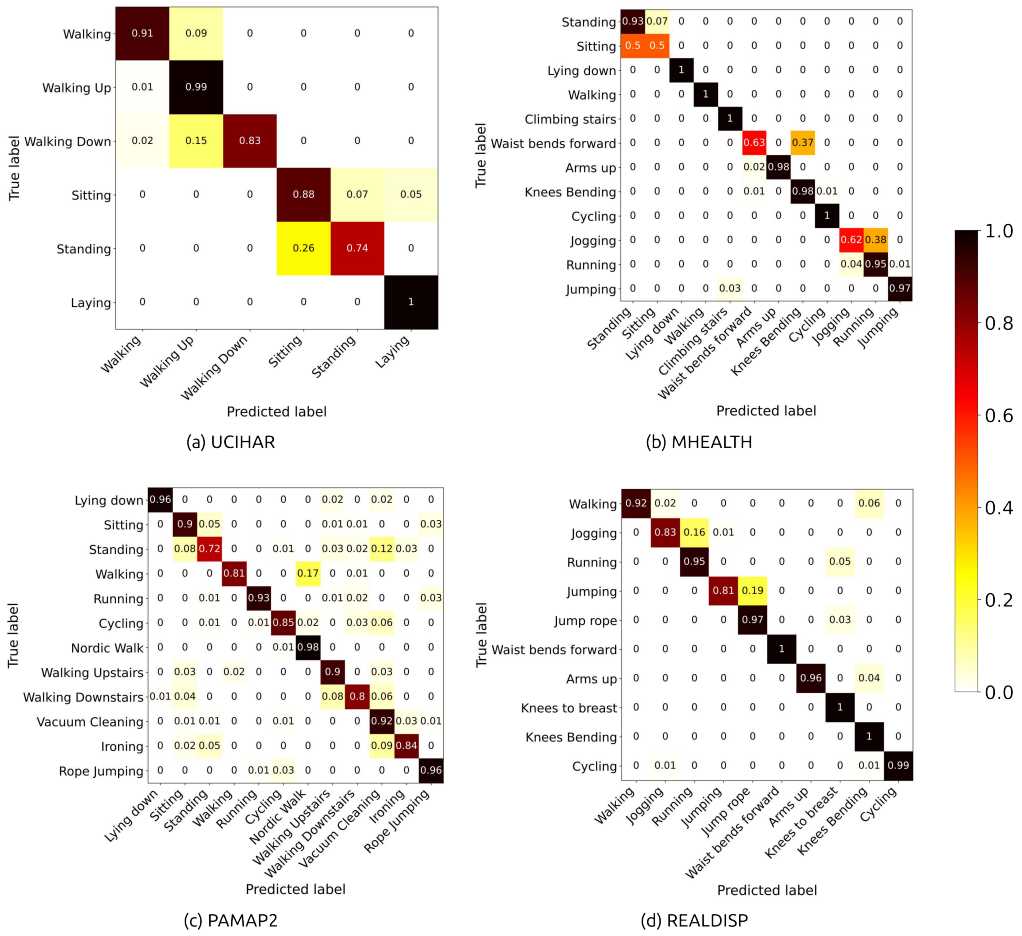


Figure 2: Confusion matrices of accuracy with the Global model on all datasets.

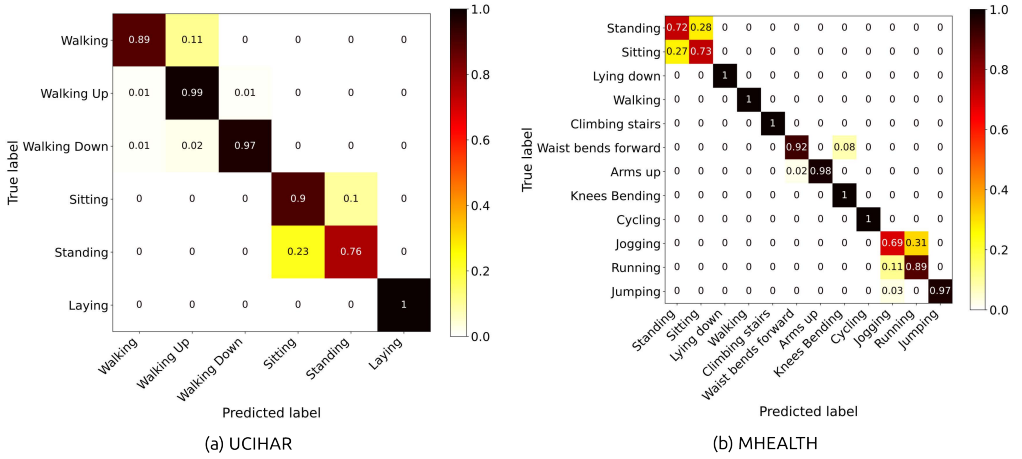


Figure 3: Confusion matrices of accuracy with the Contrastive model on UCIHAR and MHEALTH datasets.

following a contrastive learning approach. We evaluated our approach on four HAR benchmark datasets, showing a significant performance increase compared to previous studies, including CNNs, LSTMs, CNN-LSTMs, Self-Attention, and GNN-based models.

The results of the experiments show that the underlying structure of the IMU sensors' data channels and the strengths of their

connections can be modeled as a graph. In addition, they evidence that such graphs can be properly encoded by a GNN-based model and the embeddings learned by the GNNs can be used downstream for Human Activity Recognition. The results confirm that considering other types of relationships, beyond the time dimension, and incorporating spatial dependencies into the model, allows uniquely characterize each human activity. The best performing models achieved an accuracy and macro f1-score above 90% for the UCI-HAR, MHEALTH, and REALDISP datasets, and an accuracy and macro f1-score above 86% for the PAMAP2 dataset. This shows that GNNs are a good alternative for Human Activity Recognition.

A promising future direction is to combine GNNs with temporal models to fully leverage the spatio-temporal relationships and explore Temporal GNNs for HAR. Augmentation techniques to the sensor data itself can improve the contrastive learning approach further. The effects of scaling, inverting the signals, reversing the time direction, stretching and warping the time series, together with the graph augmentation techniques, are worth investigating.

References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *Proc. 21th international European symposium on artificial neural networks, computational intelligence and machine learning*. 437–442.

[2] 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.

[3] Oresti Banos, Mate Toth, and Oliver Amft. 2014. REALDISP Activity Recognition Dataset. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5GP6D.

[4] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. 2016. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems* 29 (2016).

[5] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems* 24 (2011).

[6] James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*. PMLR, 115–123.

[7] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. 2022. Investigating (re) current state-of-the-art in human activity recognition datasets. *Frontiers in Computer Science* 4 (2022), 924954.

[8] Wensong Chan, Zhiqiang Tian, and Yang Wu. 2020. Gas-gcn: Gated action-specific graph convolutional networks for skeleton-based action recognition. *Sensors* 20, 12 (2020), 3499.

[9] Yuqing Chen and Yang Xue. 2015. A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE international conference on systems, man, and cybernetics*. IEEE, 1488–1492.

[10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

[11] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. 2021. Eta prediction with graph neural networks in google maps. In *Proc. 30th ACM International Conference on Information & Knowledge Management*. 3767–3776.

[12] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* 28 (2015).

[13] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2017. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674* (2017).

[14] 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).

[15] Nils Y Hammerla and Thomas Plötz. 2015. Let's (not) stick together: pairwise similarity biases cross-validation in activity recognition. In *Proc. 2015 ACM international joint conference on pervasive and ubiquitous computing*. 1041–1051.

[16] James F Hemphill. 2003. Interpreting the magnitudes of correlation coefficients. (2003).

[17] Chenhui Hu, Lin Cheng, Jorge Sepulcre, Keith A Johnson, Georges E Fakhri, Yue M Lu, and Quanzheng Li. 2015. A spectral graph regression model for learning brain connectivity of Alzheimer's disease. *PloS one* 10, 5 (2015), e0128136.

[18] Wenbo Huang, Lei Zhang, Wenbin Gao, Fuhong Min, and Jun He. 2021. Shallow convolutional neural networks for human activity recognition using wearable sensors. *IEEE Transactions on Instrumentation and Measurement* 70 (2021), 1–11.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[21] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2021), 3316–3333.

[22] Saif Mahmud, M Tanjid Hasan Tonmoy, Kishor Kumar Bhaumik, AKM Mahbubur Rahman, M Ashraful Amin, Mohammad Shoyaib, Muhammad Asif Hossain Khan, and Amin Ahsan Ali. 2020. Human activity recognition from wearable sensor data using self-attention. In *ECAI 2020*. IOS Press, 1332–1339.

[23] Sakorn Mekruksavanich and Anuchit Jitpattanakul. 2021. Lstm networks using smartphone data for sensor-based human activity recognition in smart homes. *Sensors* 21, 5 (2021), 1636.

[24] Abduallah Mohamed, Fernando Lejarza, Stephanie Cahail, Christian Claudel, and Edison Thomaz. 2022. HAR-GCNN: Deep Graph CNNs for Human Activity Recognition From Highly Unlabeled Mobile Sensor Data. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 335–340.

[25] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4602–4609.

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

[27] Lishan Qiao, Han Zhang, Minjeong Kim, Shenghua Teng, Limei Zhang, and Dinggang Shen. 2016. Estimating functional brain networks by incorporating a modularity prior. *Neuroimage* 141 (2016), 399–407.

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

[29] Richard Taylor. 1990. Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography* 6, 1 (1990), 35–39.

[30] Andrés Tello, Victoria Degeler, and Alexander Lazovik. 2024. Too Good To Be True: accuracy overestimation in (re) current practices for Human Activity Recognition. In *IEEE Int. Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 511–517.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[32] Shaohua Wan, Lianyong Qi, Xiaolong Xu, Chao Tong, and Zonghua Gu. 2020. Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications* 25, 2 (2020), 743–755.

[33] Yan Wang, Xin Wang, Hongmei Yang, Yingrui Geng, Hongnian Yu, Ge Zheng, and Liang Liao. 2023. MhaGNN: A novel framework for wearable sensor-based human activity recognition combining multi-head attention and graph neural networks. *IEEE Transactions on Instrumentation and Measurement* (2023).

[34] Li Xiao, Aiying Zhang, Biao Cai, Julia M Stephen, Tony W Wilson, Vince D Calhoun, and Yu-Ping Wang. 2020. Correlation guided graph learning to estimate functional connectivity patterns from fMRI data. *IEEE Transactions on Biomedical Engineering* 68, 4 (2020), 1154–1165.

[35] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[36] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *32nd AAAI conference on artificial intelligence*.

[37] Yan Yan, Tianzheng Liao, Jinjin Zhao, Jiahong Wang, Liang Ma, Wei Lv, Jing Xiong, and Lei Wang. 2022. Deep transfer learning with graph neural network for sensor-based human activity recognition. *preprint arXiv:2203.07910* (2022).

[38] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems* 31 (2018).

[39] 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.

[40] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.