MaskCAE: Masked Convolutional AutoEncoder via Sensor Data Reconstruction for Self-Supervised Human Activity Recognition

Dongzhou Cheng, Lei Zhang, Lutong Qin, Shuoyuan Wang, Hao Wu and Aiguo Song, *Senior Member, IEEE*

***Abstract*— Self-supervised Human Activity Recognition (HAR) has been gradually gaining a lot of attention in ubiquitous computing community. Its current focus primarily lies in how to overcome the challenge of manually labeling complicated and intricate sensor data from wearable devices, which is often hard to interpret. However, current self-supervised algorithms encounter three main challenges: performance variability caused by data augmentations in contrastive learning paradigm, limitations imposed by traditional self-supervised models, and the computational load deployed on wearable devices by current mainstream transformer encoders. To comprehensively tackle these challenges, this paper proposes a powerful self-supervised approach for HAR from a novel perspective of denoising autoencoder, the first of its kind to explore how to reconstruct masked sensor data built on a commonly employed, well-designed, and computationally efficient fully convolutional network. Extensive experiments demonstrate that our proposed Masked Convolutional AutoEncoder (MaskCAE) outperforms current state-of-the-art algorithms in self-supervised, fully supervised, and semi-supervised situations without relying on any data augmentations, which fills the gap of masked sensor data modeling in HAR area. Visualization analyses show that our MaskCAE could effectively capture temporal semantics in time series sensor data, indicating its great potential in modeling abstracted sensor data. An actual implementation is evaluated on an embedded platform. Our code will be released at `https://github.com/cheng-haha/MaskCAE`.**

***Index Terms*— Human Activity Recognition, Sensor, Masked Reconstruction, Convolutional Autoencoder, Self-Supervised Learning.**

This work was supported in part by the National Natural Science Foundation of China under Grant 62373194, in part by the Cultivating Plan Program for the Leader in Science and Technology of Yunnan Province, China under Grant 202005AC160005, and in part by the Ten Thousand Talent Plans for Young of Yunnan Province, China under Grant YNWR-QNBJ-2019-188. *(Corresponding author: Lei Zhang. (E-mail: leizhang@njnu.edu.cn))*

Dongzhou Cheng, Lei Zhang and Lutong Qin are with the School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, 210023, China.

Shuoyuan Wang is with the Department of Computer and Information Science, University of Macau, Taipa, 999078, China.

Hao Wu is with the School of Information Science and Engineering, Yunnan University, Kunming, 650500, China.

Aiguo Song is with the School of Instrument Science and Engineering, Southeast University, Nanjing, 210096, China.

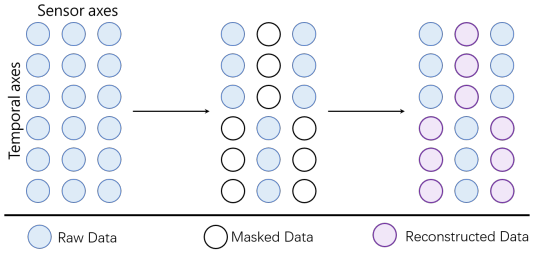


Fig. 1: Masked reconstruction along temporal and sensor axes.

I. Introduction

A. Background

The wide popularity of portable devices equipped with embedded sensors such as smartphones and smartwatches has motivated the study of sensor-based Human Activity Recognition (HAR), which becomes a spotlight in ubiquitous computing community [1]. However, it is extremely challenging to manually label time series sensor data, which is often hard to interpret than image data. To assign accurate labels, human annotators have to rely on prior experience knowledge to determine the starting and terminating locations in intricate time-series sensor data for an activity [2]. In recent years, there has been a significant interest in applying Self-Supervised Learning (SSL) to sensor data to tackle above challenge [3], [4], [5]. Specifically, self-supervised learning involves leveraging pretext tasks to learn a well-generalized feature representation on a large amount of unlabeled data, followed by fine-tuning in downstream tasks. It provides a promising solution to achieve competitive HAR algorithms that are comparable to or even surpass traditional supervised approaches, hence indicating a potential to alleviate laborious or time-consuming manual efforts and data-hungry issues [6].

B. Current challenges

In the pursuit of an improved pre-trained HAR model, prior most works usually embrace Contrastive Learning (CL) [7], i.e., a mainstream pre-training technique in self-supervised learning paradigm, which leads to significant challenges. CL methods primarily seek to learn feature representations from different views for the same sample. For instance, classical

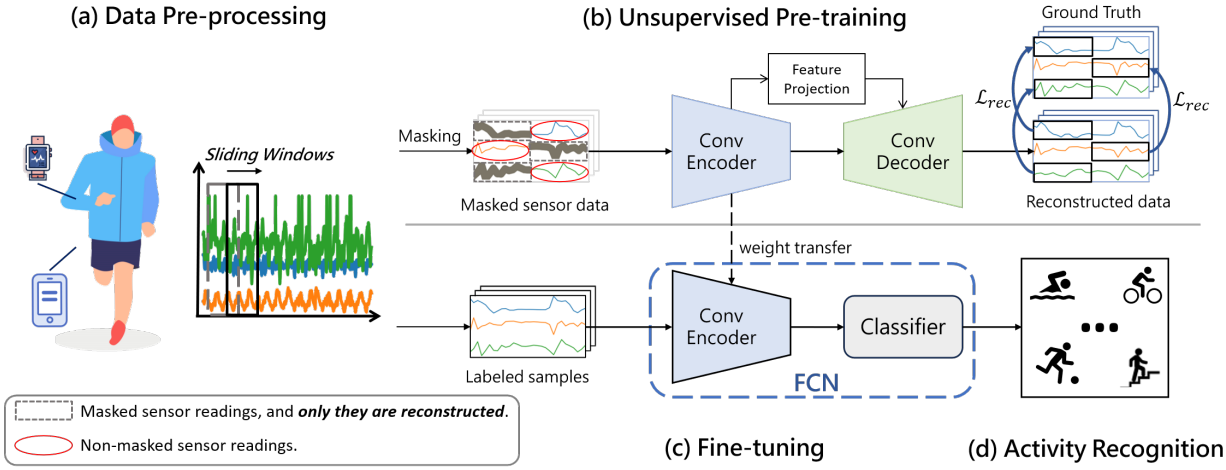


Fig. 2: An overall pipeline of MaskCAE in HAR scenario. (a) The collected sensor data first undergoes preprocessing, such as data normalization and sliding window. (b) Unsupervised pre-training involves reconstructing the mask sensor readings using the loss $\mathcal{L}_{rec}$ applied only to the mask regions. The reconstructed process enables the model to gradually capture intricate activity semantic information from ground truth. (c) After pretraining, the encoder's weights can be transferred to the downstream model followed by a classifier to output softmax probabilities for activity prediction. Then the model FCN is fine-tuned. (d) Once training is completed, the model can be deployed on embedded devices to recognize target activities.

approaches like SimCLR [8] perform data augmentation by treating different views of the same sample as positive pairs and other samples in a batch as negative pairs. However, the choices of data augmentation always play a critical role in CL, and recent literatures suggests that it might introduce substantial variance in the final performance [9], [10].

Alternatively, it is worth exploring how to apply autoregressive modeling methods as a potential SSL solution to circumvent such performance variability stemming from data augmentation [11], [12]. This strategy is simple yet effective, which utilizes an encoder to encode sensor information into a latent representation vector and employs a decoder to reconstruct the latent representation into an original input. Typically, Masked Reconstruction [3], akin to BERT [11], masks out a portion of sensor data points in the transformer [13] and reconstructs the missing semantic information using the decoder. Yet, it has been observed that the achieved performance of the self-supervised model is rather limited. Moreover, the high computational complexity of transformers poses a significant challenge when deployed on a mobile device [14].

C. Research motivation

Recent years have witnessed remarkable success of Convolutional Neural Networks (CNNs) [2], which can effectively extract hierarchical activity features from sensor data while offering fast inference speed. Therefore, to tackle the aforementioned challenges, a natural idea motivated by Masked Reconstruction arises: ***Can one take the advantage of CNN's hierarchy to reconstruct masked time series sensor data for activity inference?*** Intuitively, extending the success of self-supervised from transformers to efficient convolutional networks is a wonderful and potential. To the best of our knowledge, how to port Masked Reconstruction into efficient convolutional backbones has been rarely explored in HAR area. As we have known, representation learning for HAR is mainly based on analyzing the sensor readings extracted through a popular sliding window strategy. Similar to previous works [3], [11], [12], one can set the values across all sensor axes for these randomly chosen timesteps to zero (Fig. 1). To this end, each input window will be perturbed by a binary mask matrix with the same dimensions. Then, the final masked sensor input can be obtained by taking their dot product. This goal is to force the model to reconstruct these masked out parts, which can learn temporal patterns from context and thus make for a richer representation than that is directly derived from raw sensor data. However, when integrating CNNs with masked autoencoders for sensor data, there are two potential issues that need to be addressed. (1) One issue comes from the fact that mask modeling, that is rooted in natural language processing (NLP) tasks, usually operates in a single-scale manner [11]. Simply applying it to CNNs would inevitably result in losing the advantage of hierarchical modeling in HAR scenarios. Such hierarchical structure has always been the gold standard for HAR based on time series sensor data, which allows for substantial amplification of activity semantics information from sensor data while decreasing computational overload. Unlike Mask Reconstruction [3], that utilizes only one fully connected layer as the decoder, a straightforward solution is to employ a hierarchical decoder that may capture multi-scale encoded features from sensor signals. (2) Another issue is that unlike transformers, plain convolutional networks not only run fixed-length sliding window with overlapping but also operate on regular feature maps, which lack the capability to handle variable-length inputs from masking operations [14]. As shown in Fig. 4, directly zero-outing all masked sensor readings and feeding them into CNN might cause a severe shift in data distribution. To fill this gap, a promising solution is to first mask sensor readings randomly in a patch-wise manner. Noting that these unmasked patches can coincide well with point clouds by sharing a sparse nature, one can treat unmasked parts as a set of sparse patches and then apply sparse convolutions to handle only visible or seen

parts for encoding [15], [16], hence allowing CNN to handle irregular masked sensor input without a distribution shift. Notably, sparse convolutions can effortlessly transform into dense weights during fine-tuning, simplifying the acquisition of complete pre-trained encoder weights for CNNs.

D. Contribution

Inspired by above observations, we propose a novel approach called Masked Convolutional AutoEncoder (MaskCAE) by introducing BERT-style pre-training into a well-designed fully convolutional network for activity recognition, which in turn may be of benefit to Masked Sensor data Modeling. Fig. 2 presents an overview of data collection, unsupervised pretraining, and fine-tuning for activity inference, which utilizes a set of components including sparse convolution, feature projection, information fusion decoder, patch-level normalization mask loss, etc. to address the aforementioned challenges (Fig. 5 shows the complete framework design). In particular, sparse convolution is in charge of accurately eliminating the information of these masked parts while allowing CNNs to easily handle irregularly masked sensor inputs. For encoding, it is worth noting that signal-wise feature extractors [17], [18] act as encoders, where different sensor channels do not interfere with each other. Patch-level normalization loss with sparse convolution is used to compute only the loss of the masked portions, which ensures that the feature representation network is able to reconstruct the masked sensor signals based on the fine-grained information from the non-masked regions of sensor data. For decoding, we utilize the multi-scale decoder symmetrically with the encoder, and fill the masked embeddings to the masked parts, and then feed the encoded features to the hierarchical convolutional decoder. In particular, we highlight that the decoder need to see the contextual information among different sensor channels. In such a way, the whole sensor time-series data can be reliably reconstructed by fusing the contextual semantics from different sensor channels, even if partial sensor channels are completely masked. Extensive experiment analysis on three public benchmark datasets including USC-HAD [19], MotionSense [20], and UCI-HAR [21], validates the effectiveness of MaskCAE in fully-supervised, self-supervised, and semi-supervised scenarios. In summary, the main contributions of this paper are as follows:

- In this paper, to extend recent success of self-supervised from transformers to efficient convolutional backbones, we present a new self-supervised algorithm called MaskCAE, which utilizes masked reconstruction to model intricate time series sensor data built on a hierarchical AutoEncoder for HAR. We explore how to port Masked Reconstruction into efficient convolutional backbones. To the best of our knowledge, this paper is the first work to reconstruct masked sensor data built on efficient CNN, which fills the absence of masked reconstruction in sensor data and presents promising state-of-the-art results.
- Unlike previous most self-supervised contrastive algorithms, our approach starts from another perspective of autoregressive modeling, which does not rely on any data augmentations, hence greatly alleviating potential unstableness and performance variability caused by manual data augmentations. Further, in contrast to the transformer encoders commonly employed in the BERT-style pre-training paradigm, the full convolutional network in MaskCAE provides a reasonable and faster inference speed, which is potentially beneficial for the practical HAR deployment.
- Our method is the first use of sparse convolution and hierarchical design for masked sensor data modeling. Extensive experiments and ablation studies validate the effectiveness of the proposed method for various HAR tasks, which surpasses both state-of-the-art models by significant margins in self-supervised, semi-supervised, and fully-supervised scenarios. Interpretable visualization analyses provide a deep insight into the powerful MaskCAE. A practical implementation is evaluated on resource-limited mobile device. All these evidences reveal a promising future of masked sensor data reconstruction on hierarchical convolutional backbones.

II. Related Works

A. Human Activity Recognition (HAR)

In recent years, extensive research has been conducted in utilizing deep learning models such as RNNs, CNNs, Residual Networks, Hybrid Models, AutoEncoders, and Transformers to extract temporal features from activity sequences [17], [22], [23], [24], [25]. These studies have demonstrated outstanding performance in various activity recognition tasks. For instance, Hammerla *et al.* [25] present comprehensive experiments on RNNs, CNNs, and hybrid model DeepConvLSTM [26]. Jiang and Yin [27] tackle the HAR problem by utilizing new activity samples as inputs to deep CNNs (DCNN). To explore the optimal extraction and fusion of features from multimodal sensor data, MCNN [17] extract temporal features from each signal channel. Furthermore, Kim [18] showcase a highly effective signal-wise temporal feature extractor. However, previous literatures primarily focus on fully supervised HAR tasks. It still remains a challenge whether these works can achieve satisfactory performance when handling limited labeled data [6].

B. Self-Supervised Learning for HAR

Indeed, labeling time-series data is a challenging task, and it is impractical to assume the availability of sufficient labeled data in real-world scenarios. Self-supervised learning [7], [8], [28], originating from the realm of unsupervised learning, emerges as one of the most effective solutions to tackle this dilemma. Specifically, self-supervised learning leverages unlabeled data to learn a well-generalized representation network through pretext tasks, which is then fine-tuned for downstream tasks [8], [29]. In the context of HAR, self-supervised learning garners significant attention. Tang *et al.* [10] are the first to explore the application of contrastive learning in HAR using the SimCLR framework, employing temporal augmentations. Khaertdinov *et al.* [4] continue along the same lines, exploring

various task scenarios. Haresamudram *et al.* [30] apply a method called Contrastive Predictive Coding to HAR. Wang *et al.* [31] generated soft labels by using an unsupervised clustering method to mask negative samples of the same cluster. Qu *et al.* [9] study different contrastive learning algorithms on HAR datasets and provide detailed ablation experiments. However, it's worth noting that the current works in contrastive learning often rely on data augmentation, inevitably increasing the complexity of the task. Finding suitable data augmentations remains a question to be addressed [9], [10]. Moving on to the domain of HAR based on autoregressive modeling, Masked Reconstruction [3] explores how to reconstruct masked sensor data on transformer [13]. CAE [32] employs convolutional models as both an encoder and a decoder, reconstructing complete signal inputs using latent representation vectors. However, previous works cause slow inference on Transformer-based backbones, which is not suitable for resource-limited mobile devices. Therefore, it deserves further investigation on how to port masked reconstruction into full convolutional networks for a better accuracy-cost trade-off in self-supervised HAR scenarios. Woo *et al.* [33] and Tian *et al.* [34] have first introduced the use of sparse CNN for masked self-supervision on computer vision tasks. Different from [15], [33], [34] targeting at vision research community, this paper is mainly centered on the advancement and refinement of SSL for HAR based on multimodal sensor data, which has been rarely explored in this area. This spurs us to take a deeper look at the fundamental difference between image and sensor data processing. The primary objective of the study is to address one crucial issue present in existing HAR methods [2], [3], [6], [9], [10], [30], [32], which are dealing with annotation scarcity in sensor data by randomly masking out missing sensor readings and constructing an effective pre-training task, while trying to apply previous transformer-specialized SSL algorithm to pure convolutional backbone for HAR. We introduce MaskCAE, which tackles above issue by designing a hierarchical encoder-decoder structure along with temporal masking strategy. It primarily leverages the temporal correlations to discover the signal fluctuations in multimodal sensor data, so as to achieve superior results while relieving the shortage of labeled sensor data.

C. Autoregressive Modeling

The Denoising Autoencoders (DAE) [35], [36] are a type of autoencoder that uses a corrupted version of a signal as input to reconstruct the uncorrupted input signal. Masked Language Modeling (MLM) [11] has seen great success in natural language processing. Inspired by it, BEiT [37] brought the masked prediction paradigm to computer vision and showed the potential of Masked Image Modeling (MIM) on various tasks [36]. MAE [12] proposed predicting original pixel values for masked image patches. Prior most Masked Modeling literatures have focused on language and visual tasks, and there has been less efforts in handling sensor data. To the best of our knowledge, this paper is the first work in masked sensor data reconstruction built on efficient convolution network backbones, aiming at striking a better accuracy-latency trade-off for on-device activity inference.

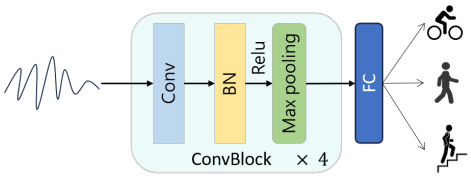


Fig. 3: Fully Convolutional Network (FCN) for HAR. *Conv* denotes the convolutional layer, and *BN* denotes Batch Normalization. *Max pooling* down-samples the temporal features. *FC* denotes fully connected layer.

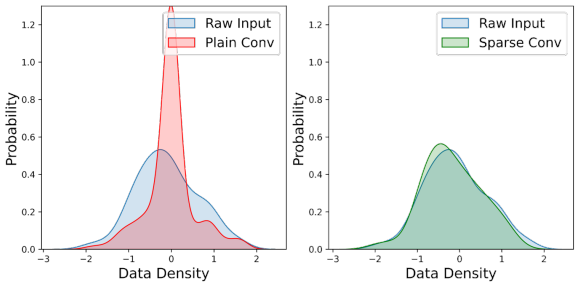


Fig. 4: Data density on a random sample from USC-HAD. Masked sensor data is assigned a value of 0. If plain convolution is applied without skipping the masked region, the convolutional operation will see a large number of 0 values, severely compromising the integrity of sensor features. Sparse convolution, on the other hand, directly skips the masked area, resulting in a data distribution similar to the original input. This capability allows the model to retain sufficient activity semantic information.

III. Method

A. Three Guidelines for Designing Lightweight Convolutional HAR Encoder

In this section, based on previous mainstream HAR literatures [1], [2], we design the fully convolutional network (FCN) according to the following three basic guidelines:

- **G1**: **Local activity features of temporal sensor data should be extracted.** In time-series sensor data, noise needs to be filtered, and not every sensor reading is useful. For HAR tasks, the extraction of local features is particularly crucial [17], [26], [27], [38].
- **G2**: **Different sensor channels should not interfere with each other.** Independently extracting temporal features from each sensor channel can be an ideal solution [17], [18], [26], because merging or fusing unrelated sensor modalities might potentially deteriorate recognition performance [22], [39].
- **G3**: **The temporal receptive field of activity features should be properly increased.** Too small receptive field could not capture long-range or global dependencies in sensor data, which causes the model to lose temporal semantic information [40]. Intuitively, non-parametric subsampling such as pooling operation can alleviate this issue while reducing computational overhead [38], [40].

Fig. 3 presents an overall design of FCN, and Table. I introduces the detailed configuration of our FCN. According to the three guiding principles, we design such hierarchical FCN

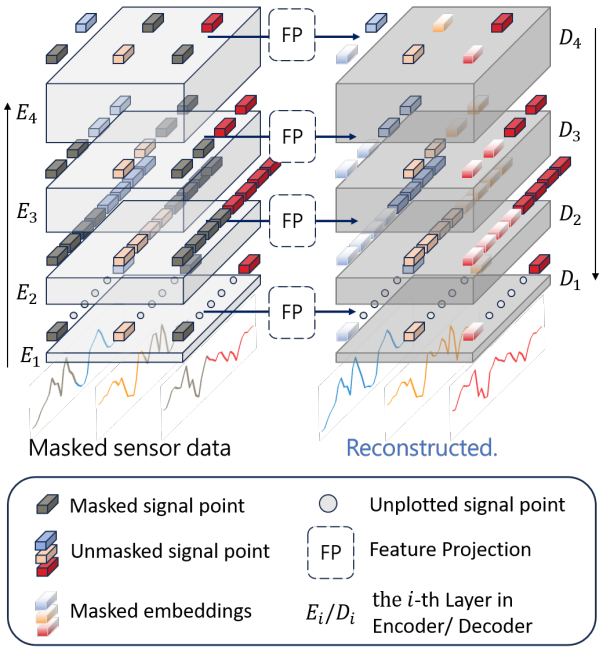


Fig. 5: The MaskCAE framework.

as our Encoder for MaskCAE, which can offer a powerful backbone to capture discriminative feature representations. First, to extract robust local features, we perform convolutional operation followed by batch normalization (BN) and nonlinear activation (ReLU) to filter out irrelevant disrupted information while extracting discriminative activity features, which can well conform to **G1**, because convolutional kernel can be competent in capturing the local relevance of time series sensor data, and the translational invariance introduced by locality; Second, a $k * 1$ convolutional kernel is slide over sensor data to capture temporal activity features. Here, we employ one-dimensional (1D) convolution over each individual univariate time series signals for temporal feature extraction during encoding phase. While there were multiple sensors axes, multivariate time series may be produced, thereby requiring such 1D convolution to be performed separately. This is consistent with **G2**; Third, if a large time scale is desirable, a max-pooling is applied between two successive CNN layers for subsampling. In light of hierarchical design, such layer-by-layer structure can allow FCN to model high-level abstractions from intricate sensor data in a multi-scale way, which is in well line with **G3**.

B. Masked Sensing Modeling

Our approach is conceptually straightforward and can operate over the aforementioned FCN backbone. The feature representations can be learned by randomly masking [12] raw continuous sensor data points at a certain rate. Then the model predicts these masked unseen portions based on the provided contextual information. The whole framework is shown in Fig. 5, and relevant details are described as follows:

Sparse Encoder design. Previous popular Masked AutoEncoder (MAE) approaches [3], [11], [12] have been primarily built on transformer-based backbones, focusing on using the decoder to reconstruct patch-level information that is missing in the encoder. It is worth noting that during encoding phase, only non-masked patches are used for information transfer. As a consequence, it will be extremely challenging while port masked reconstruction into standard convolutional networks, Unlike Transformers, convolutional backbones usually only perform sliding window over regular grid, which lacks an enough ability to handle sensor data with variable input lengths. As shown in Fig. 4, successively applying such regular convolution might potentially erode masked regions (zero positions), causing a severe data distribution shift [33], [34]. Without loss of generality, sparse convolution is employed to address this issue [15], [16], which allows us to skip the masked parts on sparse feature maps and only calculates at unmasked positions. Specific details are introduced as follows. Given raw sensor input $x \in \mathbb{R}^{T \times S}$ (T and S denote temporal and sensor dimensions respectively), the masked sensor sample may be formally defined as $x^m = x \odot \mathrm{M}_0$, where $\odot$ is the hadamard product. The binary mask matrix $\mathrm{M}_0 \in [0,1]^{T \times S}$ can be randomly generated with the values of 1 and 0, which denote the corresponding unmasked and masked regions. As shown in Fig. 5, the N-layer FCN backbone is used as our enconder $\mathcal{E} = E_1 \circ E_2 \circ \cdots \circ E_{\mathrm{N}}(\cdot)$, where E_i is the i-th Encoder layer and $\forall i \in \{1 \leq i \leq N = 4\}$, which is in charge of generating a series of sparse feature maps $\{F_{E_i}\}_{i=1}^{\mathrm{N}}$ by ConvBlock$^{\mathrm{Sparse}}$ (i.e., Sparse Convolution) [15] at different scales. Here sparse convolution means that it only computes while the kernel center lies at a non-maksed position. During pre-training procedure, we encode the masked sensor data in a layer-by-layer manner:

$$\begin{aligned} F_{E_0} &= x^m, \\ F_{E_i} &= \mathrm{ConvBlock}_i^{\mathrm{Sparse}}(F_{E_{i-1}}) \\ &= E_i\left(F_{E_{i-1}}\right) \odot \mathrm{M}_i, \quad \forall i \in [1,2,3,4], \mathrm{M}_i \in \mathbb{R}^{\frac{T}{2^i} \times S}, \end{aligned} \tag{1}$$

where $E_i(\cdot) \odot \mathrm{M}_i$ denotes sparse operation that removes the masked sensor features by mask matrix $\{\mathrm{M}_i\}_{i=1}^{\mathrm{N}}$. Taking our FCN backbone as an example, each of $\{E_i\}_{i=1}^{\mathrm{N}}$ is followed by a subsampling operation. Keeping the sensor dimension unchanged, the sparse feature maps F_{E_i} will be successively subsampled by a factor of 2 after every block. For an input sensor window, the encoder can produce feature maps $\{F_{E_i}\}_{i=1}^{\mathrm{N}}$ at 4 temporal scales with tensor shapes of $\{\frac{T}{2} \times S, \frac{T}{4} \times S, \frac{T}{8} \times S, \frac{T}{16} \times S\}$ (also see Table. I). As illustrated in Fig. 5, even as the temporal features are halved at each level, the masked positions remain unchanged (with black squares representing masked data points). This scheme can help to avoid the deformation of mask pattern caused by convolution, while maintaining a consistent masking ratio through all convolutional layers.

It's worth noting that the encoder uses an FCN as the backbone network. During pre-training phase, sparse convolutions force the model to only see unmasked sensor data points, thereby encouraging the encoder to learn how to reconstruct these unseen masked positions. Importantly, the sparse convolutional layers are transformed back into standard dense convolutions during fine-tuning stage (*Pre-training:*$E_i(\cdot) \odot \mathrm{M}_i \rightarrow$ *Evaluation:*$E_i(\cdot)$), which could seamlessly transfer the model weights to the FCN with the same size.

Information Fusion Decoder design. MaskCAE uses the UNet [41] module as its decoder. However, if the UNet [41] module only extracts features along the temporal axis in an independent or separate manner (**G2**), it may fail to capture contextual semantics between different sensor channels, making it challenging to reconstruct the missing information that matches the ground truth. A simple yet effective solution is to apply square kernel convolutions (i.e., kernel size: 5×5) to fuse information from different sensor channels, which can allow for better decoding of latent features from the encoder. We explain why the main difference occurs between Encoder and Decoder. During encoding phase, one main concern is that there might exist the temporal disparity across different sensing modalities (e.g., different sensors axes, or channels). As a result, directly merging or fusing unrelated sensor modalities might potentially deteriorate model performance. To avoid temporal disparity, one should extract time features independently from each individual univariate sensor channel by separately applying $k \times 1$ convolution kernel to slide along the time axis (**G2**). In other words, the Encoder's duty is to discover the signal fluctuations within a fixed temporal range, and then transfer the Encoder's weights to FCN to capture as much as possible temporal feature representation. Instead, after reshaping original sensor data with the Encoder, the Decoder is in charge of hierarchically receiving masked sensor data, which aims to strengthen the Encoder's sequential activity modeling by reconstructing these missing sensor readings. Thus, for reconstruction purpose, an ideal solution is to use $k \times k$ convolution kernel, which enable the Decoder to see nearby data points across different sensor axes or channels. In such a way, more activity semantic information can be captured to better reconstruct the corresponding masked regions. Before reconstructing, it's essential to map the information from each level of the encoder to the decoder using Feature Projections (FP). Without loss of generality, as shown in Fig. 5, assuming that E_i and D_i denote the i-th modules of the encoder and decoder, respectively, the features at the i-th level can be mathematically formulated as F_{E_i} and F_{D_i}:

$$\begin{aligned} F_{D_4} &= \mathrm{FP}_4\left(F'_{E_4}\right), \\ F_{D_i} &= D_i\left(F_{D_{i+1}}\right) + \mathrm{FP}_i\left(F'_{E_i}\right) \quad (\forall i \in \{3,2,1\}). \end{aligned} \tag{2}$$

Specifically, F'_{E_i} represents the masked sensor data points, where these masked embeddings $[\mathrm{M}_i]$ (as shown in Fig. 5) can be inserted into the empty positions over sparse feature maps, which are then mapped back to the decoder by using FPs (i.e., a single convolutional layer without downsampling). Later ablation studies will validate the effectiveness of these modifications.

Reconstruction target. According to the guiding principle from previous MAE literatures [12], [33], we choose the Patch-level Normalization L2-Loss [12] as L_{rec} to pull the reconstructed sensor signal close to the target sample:

$$L_{rec} = \left\| (x - F_{D_1}) \circ (1 - \mathrm{M}_0) \right\|^2, \tag{3}$$

where MaskCAE only calculates the loss at the masked positions, so as to reduce the loss from a matrix to a scalar. After pretraining, the decoder is discarded and the pre-trained encoder's weights can be transferred to downstream tasks.

TABLE I: Lightweight FCN Settings. ConvBlock (CB) is shown in Fig. 3.

Stage	#Features	Layer Specification		Settings
Encoder				
1	$\frac{T}{2} \times S$	CB	channel dim	16
			kernel size	5*1, stride (1,1)
			pooling size	5*1, stride (2,1)
2	$\frac{T}{4} \times S$	CB	channel dim	32
			kernel size	5*1, stride (1,1)
			pooling size	5*1, stride (2,1)
3	$\frac{T}{8} \times S$	CB	channel dim	64
			kernel size	5*1, stride (1,1)
			pooling size	5*1, stride (2,1)
4	$\frac{T}{16} \times S$	CB	channel size	128
			kernel size	5*1, stride (1,1)
			pooling size	5*1, stride (2,1)
Classifier				
Flatten, FC layer				

TABLE II: Statistical information of datasets and experiment setup. *#Sensors* represents the type of sensor channels used (A = accelerometer and G = gyroscope). *#SW* denotes the sliding window length.

Dataset	USC-HAD	MotionSense	UCI-HAR
#Sensors	A,G	A,G	A,G
#Subjects	14	24	30
#Class	6	12	6
Freq (Hz)	30	30	30
#SW	1 s	1 s	1 s
#Class	6	12	6
#Train samples	35608	37265	28096
#Valid samples	9201	7013	5364
#Test samples	10514	11709	7736
Pre-training		Linear probing or Full fine-tuning	
Optimer	LAMB [42]	AdamW [43]	
Learning rate	2e-3	1e-4	
Epochs	1000	100	
Batch size	512	512	
Mask ratio	0.3	-	

IV. Experiments

A. Experimental setting

Benchmark datasets. In line with prior most studies [4], [3], [30], we select three of the most commonly employed benchmark datasets for performance evaluation: USC-HAD, MotionSense, and UCI-HAR, whose details are introduced as follows. **USC-HAD** [19]: This dataset contains sensor readings recorded from 14 subjects, who perform 12 different activities such as 'forward walking,' 'left walking,' 'right walking,' 'jumping,' 'sitting,' 'standing,' and 'sleeping.' **MotionSense** [20]: This dataset involves 24 participants with varying genders, ages, weights, and heights, where each person carrying an iPhone 6s engages in six types of activities: 'descending stairs,' 'ascending stairs,' 'walking,' 'jogging,' 'sitting,' and 'standing.' **UCI-HAR** [21]: This dataset is composed of sensor recordings from 30 subjects, who are instructed to perform six kinds of activities of daily living (ADLs): 'standing,' 'lying,' 'sitting,' 'walking,' 'walking upstairs,' and 'walking downstairs.' Each subject wears a waist-mounted smartphone (Samsung Galaxy S II) equipped with embedded inertial sensors. Following the same data splitting protocol as previous literatures [3], [4], [30], UCI-HAR and MotionSense utilize 20% of subjects for testing, while holding out the 20% of remaining subjects for validating. USC-HAD utilizes subjects 11, 12 for validation and subjects 13, 14 for testing,

TABLE III: Self-supervised learning results (Bold font highlights the fully-supervised FCN only, and the two different self-supervised settings, i.e., linear probing and full fine-tuning for our MaskCAE). The methods in the table are assumed to use Linear Probing (LP) by default. † denotes Full Fine-Tuning (FFT). CL and AM are *Contrastive Learning* and *Autoregressive Modeling* in Self-Supervised Learning (SSL) algorithms, respectively.

Method	Type	Backbone	USC-HAD	MotionSense	UCI-HAR
1D Conv [30]	Sup		49.09	86.66	79.79
DeepConvLSTM [26]	Sup		44.83	85.15	82.83
Transformer [3]	Sup		43.84	83.30	82.61
CNN-Transformer [4]	Sup		60.56	-	95.26
FCN Only	Sup		**60.13**	**89.43**	**93.66**
Multi-task SSL [5]	SSL	TPN	45.37	83.30	80.20
CPCHAR [30]	CL	1D Conv	52.01	89.05	81.65
CSSHAR [4]	CL	CNN-Transformer	57.76	-	91.14
Masked Reconstruction [3]	AM	Transformer	49.31	88.02	81.89
CAE [32]	AM	CNN	48.82	82.50	80.26
MaskCAE (LP)	AM	FCN	**57.36**	**89.64**	**93.19**
MaskCAE (Ours)†	AM	FCN	**64.32**	**90.35**	**94.59**

while holding out the rest for training. For fair comparisons, the original accelerometer and gyroscope signals are further downsampled into 30Hz and divided into 1-second time windows with 50% overlap. Table. II summarizes the specific details. We perform five runs to report their average values as final results.

Evaluation Metrics. Due to the potential issue of class imbalance in HAR situations, following previous literatures [4], [30], we choose the averaged F1-score as the primary performance metric:

$$F_1 - score = \frac{1}{C} \times \sum_{i=1}^{C} \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \tag{4}$$

where C represents the total number of classes. Precision_i and Recall_i for the i^{th} activity class are defined as $TP_i/(TP_i + FP_i)$ and $TP_i/(TP_i + FN_i)$, respectively. Here, TP_i, FP_i, TN_i, and FN_i denote true positives, false positives, true negatives, and false negatives in the i^{th} activity class.

Implementation details. Different from previous most contrastive learning works [4], [9], [30], we do not apply any data augmentation to original sensor data. During the pre-training phase, a LAMB [42] optimizer is employed to train the Encoder-Decoder architecture at a masking ratio of 0.3. The training process lasts 1000 epochs with a batch size of 512, while the cosine annealing learning rate is $2e-3 * batchsize/256$. During the fine-tuning phase, we transfer the encoder's weights to the FCN backbone (Table. I). An AdamW [43] optimizer is utilized to fine-tune the FCN for 100 epochs with a cosine annealing learning rate $1e-4 * batchsize/256$. In addition, following the same strategy in previous works [3], [12], [31], the encoder is first pre-trained on unlabeled training data. Without loss of generality, we evaluate two most popular fine-tuning protocols [12] by adapting the pre-trained models on the whole training set with the original activity labels: Linear Probing (updating only the last linear classification layer) and Full Fine-Tuning (updating all parameters of the pre-trained model). Specific details are summarized in Table. II. Unless otherwise specified, these implementational details are considered as default settings.

B. Self-Supervised Learning

In practice, the most common strategy for comparing representations learned by self-supervised methods is to first do pre-training without using class labels, and then use the learned representations for fine-tuning downstream tasks such as human activity recognition. This strategy has been widely employed to evaluate the quality of feature representations learned by quantifying performance [44]. Therefore, similar to previous literatures [3], [4], [5], [30], [32], in Table. III, we primarily compare the performance of our MaskCAE approach to state-of-the-art SSL techniques, e.g., Multi-task self-supervision [5] and Convolutional Autoencoder (CAE) [32] (and, for reference only, to supervised learning pipelines such as DeepConvLSTM [26] and Transformer classifier [3]), given that the main focus of this paper is on the self-supervised pretraining-finetuning paradigm. To observe the performance ceilings that the learned feature representations can attain, here we first report only the fully-supervised FCN F1-score. Built on the FCN backbone, then we consider two different self-supervised settings: MaskCAE by linear probing and MaskCAE by full fine-tuning, which are the aforementioned two standard fine-tuning protocols [12], [33]. We pretrain models using the proposed MaskCAE framework, and compare the full fine-tuning results to the linear probing counterparts. That is to say, we gradually apply the two most important designs (FCN and MaskCAE) in our proposed MaskCAE framework (also see Fig. 2) and check the corresponding performance improvements respectively. Following the linear probing setup used in previous work [3], [4], [30], we find that MaskCAE's performance can be very close to that of fully supervised FCN on various datasets. It's worth noting that our approach outperforms state-of-the-art autoregressive modeling methods by significant margins of 8.05%, 1.62%, and 11.30% on USC-HAD, MotionSense, and UCI-HAR datasets, respectively. Furthermore, MaskCAE without data augmentation is on par with self-supervised contrastive learning baselines, but can perform in a more simplified way without being plagued by variable manual data augmentations. With the full fine-tuning strategy, it can be seen that the pre-trained feature representations obtained through MaskCAE offer better network initializations under full supervision, which yield 4.19%, 0.92%, and 0.93%

TABLE IV: Comparing our proposed MaskCAE with several existing state-of-the-art SSL methods based on various data augmentations. None denotes no data augmentation.

Data Augmentation	SSL Methods			
	SimCLR	BYOL	SimSiam	TS-TCC
noise	93.64	94.17	83.93	93.40
scale	90.00	88.16	82.91	89.17
negate	81.94	84.95	80.97	90.34
perm	88.25	90.73	85.63	90.29
shuffle	74.90	83.20	80.78	92.14
resample	91.84	91.7	88.25	91.65
rotation	54.08	88.2	87.57	89.61
perm+noise	89.51	89.90	81.41	93.79
scale+noise	93.54	89.08	81.02	90.97
	MaskCAE(Ours)			
None	**94.29**			

TABLE V: Task Complexity. CL and AM are defined in Table. III. As n data augmentations are applied to CL methods, the overall performance complexity will tend to $O(n)$.

	CL	AM
complexity	$O(n)$	$O(1)$

performance gains over FCN on USC-HAD, MotionSense, and UCI-HAR datasets, respectively. In summary, it can be seen that our MaskCAE exhibits the highest improvements over fully-supervised FCN baselines in the table, and validate that both hierarchical design and sparse masking strategy are promising.

Obviously, MaskCAE appears to be an ideal solution in the realm of self-supervised HAR. Following previous work that explores the effectiveness of contrastive learning in HAR area [9], we split the whole UCI-HAR dataset into training set, validation set, and test set according to a ratio of 64%, 16%, and 20%. As shown in Table. IV, we empirically compare our proposed MaskCAE with the existing state-of-the-art SSL methods based on data augmentation [9], [10]. One can clearly observe the performance variations caused by different data augmentation techniques, where there is no unique augmentation transformation that can consistently perform better than others in all cases. The classification results of existing most SSL methods drastically vary, suggesting that automatically finding a suitable data augmentation solution for contrastive learning will be extremely challenging, which still remains an open problem [9]. In contrast, our proposed MaskCAE does not apply any data augmentation to raw sensor input, which instead seeks to strengthen activity feature representation by reconstructing these masked sensor readings. In such a way, it could effectively mitigate performance variations caused by manual data augmentations, while avoiding the laboring or time-consuming human intervention. Therefore, comparing to previous works [9], [10], our proposed MaskCAE has an obvious advantage, as it does not rely on sophisticated augmentations that have proven to be essential for contrastive learning. From Table. V, it is evident that MaskCAE outperforms different contrastive learning baselines in terms of complexity, which strongly supports our research motivation.

TABLE VI: Comparisons with fully-supervised learning results (**bold** font highlights the best results). Latency is tested on a *Raspberry Pi 4* on ARM architecture with batch size of 1. Att. Model [23] is re-implemented according to the original paper, while TransFormer. HAR [9] are based on the publicly available official code. To ensure a fair comparison, we follow the same experimental setup as [20], [45] for UCI-HAR and MotionSense (specifically, a time window of 2.56 seconds), keeping USC-HAD consistent with Table. II in [24], [46].

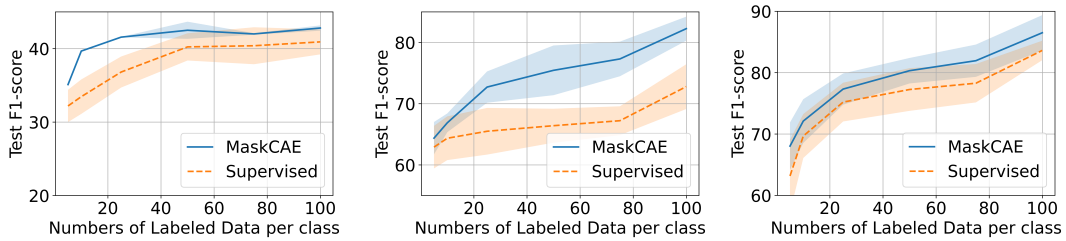
Model	Params (M)	FLOPs (M)	Latency (ms)	Acc (%)	F1 (%)
Based on USC-HAD					
Att. Model	0.475	24.92	36.62	54.31	49.30
TransFormer. HAR	0.333	20.62	34.11	50.73	47.86
FCN	0.063	1.83	3.89	64.34	60.13
MaskCAE	**0.063**	**1.83**	**3.89**	**68.48**	**64.32**
Mahmud *et al.* [24]	-	-	-	-	55.00
Khaertdinov *et al.* [46]	-	-	-	-	62.80
Based on MotionSense					
Att. Model	0.671	43.80	223.50	92.23	90.19
TransFormer. HAR	0.333	85.99	57.49	90.88	87.81
FCN	0.091	14.57	9.49	95.37	95.28
MaskCAE	**0.091**	**14.57**	**9.49**	**96.83**	**96.24**
Malekzadeh *et al.* [20]	-	-	-	96.20	95.90
Zhang *et al.* [47]	-	-	-	-	95.66
Based on UCI-HAR					
Att. Model	0.572	34.36	218.50	91.94	91.01
TransFormer. HAR	0.333	85.89	57.74	91.09	91.16
FCN	0.082	10.93	8.60	95.16	95.28
MaskCAE	**0.082**	**10.93**	**8.60**	**97.23**	97.35
Jiang *et al.* [27]	-	-	-	95.18	-
Liu *et al.* [45]	-	-	-	97.10	**97.40**

C. Fully Supervised Learning

In this section, we explore the differences in inference time and performance between the current Attention-based HAR models (Hybrid Model: Att. Model [23]; Pure Transformer Model: TransF. HAR [9]) and MaskCAE. Furthermore, we compare them to current state-of-the-art models such as SA-HAR (Mahmud *et al.* [24]), T-WaveNet (Liu *et al.* [45]), IF-ConvTransformer (Zhang *et al.* [47]), DCNN (Jiang *et al.* [27]). It can be observed that MaskCAE consistently outperforms or approaches previous state-of-the-art methods in terms of five metrics: Params, FLOPs, Latency, Accuracy, and F1. Thanks to the well-designed guidelines (**G1**, **G2**, **G3**), FCN that fine-tunes the weights by the MaskCAE's encoder can provide faster inference and superior performance compared to attention-based models and hybrid models. The failure of attention-based HAR models may be attributed to the small dataset size, which hinders the modeling of global dependencies while lacking local Inductive Biases. Experimental results also suggest an enhancement in feature representation quality in case of longer time windows (e.g., 2.56s in UCI-HAR). Overall, MaskCAE may be a well-generalized solution that achieves a better balance between inference time and performance.

D. Semi-Supervised Learning

To investigate the performance of representations trained through MaskCAE across different sample quantities, we conduct experiments in a more realistic semi-supervised setting following [3], [30]. Without loss of generality, we evaluate the fine-tuned MaskCAE and the Baseline (end-to-end trained FCN) on labeled samples $x \in$ [5, 10, 25, 50, 75, 100] for each class. The averaged F1-score from five runs is reported



(a) USC-HAD (b) MotionSense (c) UCI-HAR

Fig. 6: Semi-Supervised Learning results.

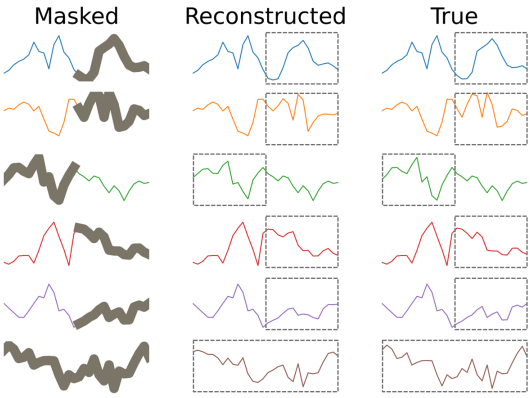


Fig. 7: Visualizing reconstructed sensor signals.

TABLE VII: Main ablations.

Method	F1	↓
Supervised	*60.13*	-
MaskCAE	**64.32**	**0.0**
w/o masking	60.69	-3.63
w/o sparse conv	62.34	-1.78
w/ plain decoder	60.90	-3.42
w/o FeatProj	63.03	-1.29
w/o norm loss	62.99	-1.33

in Fig. 6. Notably, on each dataset, MaskCAE consistently outperforms Baseline when there are few labeled samples per class. For instance, in the case of USC-HAD with only 25 labeled samples per class, MaskCAE shows a performance improvement of 6.3%. When keeping 50 labeled samples per class on MotionSense, MaskCAE is around 10% higher than Baseline. As the labeled sample quantity increases, MaskCAE continues to maintain its leading position, suggesting that it can work as a robust feature classifier.

E. Visualization

Next, we present three activity samples (i.e., Masked, Reconstructed, and Ground Truth) randomly selected from the USC-HAD validation set to visually check whether our MaskCAE can perform well in masked sensor data reconstruction. To the best of our knowledge, this is the first showcase of masked reconstruction in HAR research community. As shown in Fig. 7 , several interesting regions are highlighted by dashed rectangles, which represent the reconstructed sensor signals compared to raw sensor input. To provide a better visualization for interpretability, we randomly select an activity sample to arrange accelerometer x, y, z axes, and gyroscope x, y, z axes in a sequential manner. It can be seen that MaskCAE fits raw sensor input quite well within these masked positions, which can perfectly capture the semantics information at low or medium frequencies. Though there has been a slight mismatch at some fine-grained data points, the model is still able to make consistent and these masked positions (e.g., in the bottom line). Therefore, we believe our MaskCAE can make masked modeling well-suited for continuous sensor signals, leading to a performance leap on downstream HAR tasks.

V. Ablations

A. The role of different components

To evaluate the effectiveness of each independent component in MaskCAE, Table. VII provides a detailed ablation analysis through its five variants on USC-HAD dataset while removing one or multiple components. Overall, each component plays a crucial role in MaskCAE. 'Supervised' denotes the Fully Supervised FCN training. 'w/o masking' indicates that one only employs MaskCAE to reconstruct raw sensor input similar to CAE, rather than predicting masked sensor data. The performance degradation caused by this variant validates the effectiveness of masked sensor data modeling in embracing self-supervised paradigm, thus establishing its core contribution in HAR task. 'w/o sparse conv' indicates one does not use sparse convolution in the encoder, which forces the encoder to see these non-masked and masked positions at the same time by applying regular convolution during pretraining. Obviously, this is a suboptimal strategy to make the encoder see masked sensor data, leading to a potential data distribution shift. 'w/ plain decoder' indicates that a plain decoder only extracts activity features along temporal axis in a separate manner. However, this will cause a severe information loss of contextual semantics, thereby validating the necessity of the Information Fusion Decoder that considers different sensor channels. 'w/o FeatProj' indicates that one does not apply the feature projection branch while preserving masked embeddings. The performance gap brought by this variant suggests the importance of information interaction between the encoder and decoder. 'w/o norm loss' indicates a significant performance drop while removing normalization loss, suggesting that normalizing the loss can help to capture more latent features and improve representation power of our MaskCAE.

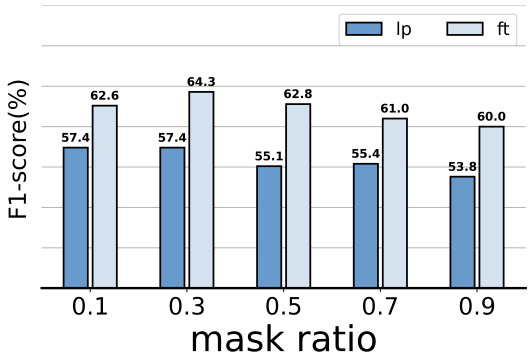


Fig. 8: Performance comparisons with different mask ratios, where 'lp' is linear probing while 'ft' denotes full fine-tuning.

B. Mask ratios

Fig. 8 analyzes the influence of mask ratio on MaskCAE. It can be observed that MaskCAE offers an acceptable mask range between [0.1, 0.7] in both linear probing and full fine-tuning evaluations, where the optimal mask ratio is close to 0.3. In addition, Fig. 9 visualizes the reconstructed results in terms of different mask ratios. By highlighting the masked parts with dashed boxes (green and purple curves indicating the z-axial acceleration and y-axial gyroscope signals, respectively), we can see that as the mask ratio increases, the reconstructed signals gradually deviate from the ground truth, which starts to lose the temporal contextual semantics of target activities. All these evidences reveal that a higher mask ratio might lead to information loss, which can be attributed to the fact that temporal sensor data does not possess as much spatial redundancy. Unlike image data, two adjacent sensor readings may be highly correlated due to their continuous nature in raw sensor signals. Therefore, our mask ratio is relatively lower compared to image-based MAE [12]. Intuitively, temporal sensor signals have a high resemblance to language data, which are artificially generated, highly semantic, and information-dense. A higher mask ratio would have a negative effect on information density, resulting in the loss of temporal contextual semantics and the failure of masked modeling.

C. Partial fine-tuning.

During recent years, linear probing has been a popular evaluation protocol in self-supervised HAR. However, this strategy will miss the opportunity to exploit stronger non-linear features, i.e., an obvious strength of deep learning. As a middle comprise, we follow earlier work [12] to add a partial fine-tuning protocol to check how performance is affected if one fine-tunes only the last few layers while freezing the remaining layers. Without loss of generality, we compare linear probing, partial fine-tuning, and full fine-tuning on USC-HAD. As shown in Fig. 10, it can be seen that linear probing (i.e., 0-layer) performs the worst among them. All partial fine-tuning results are obviously better than linear probing. We observe an interesting phenomenon: only fine-tuning a few layers can gradually improve F1 score close to full fine-tuning. Contrast to linear probing, full fine-tuning can significantly boost the F1 score from 57.4% to 64.3%, which can be attributed to stronger nonlinear features. In this paper, unless otherwise specified, we apply full fine-tuning as default setting.

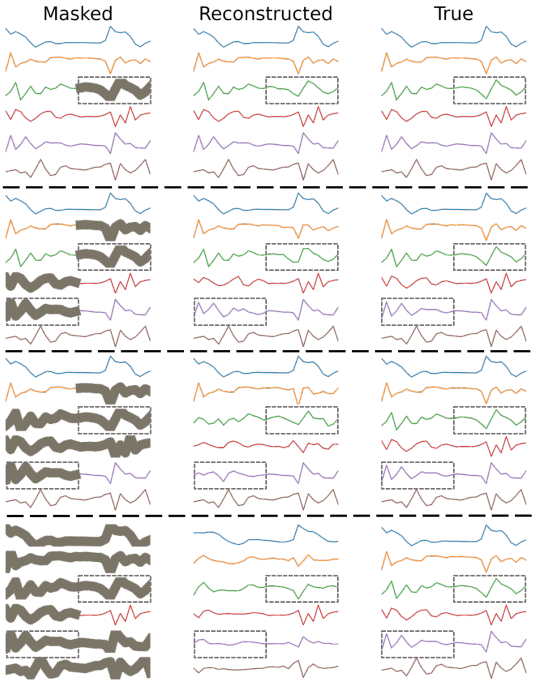


Fig. 9: Visualization results at different mask ratios for an activity sample of "Jumping Up". The first, second, third, and fourth rows represent the reconstructed results at mask ratios of 0.1, 0.3, 0.5, and 0.9, respectively.

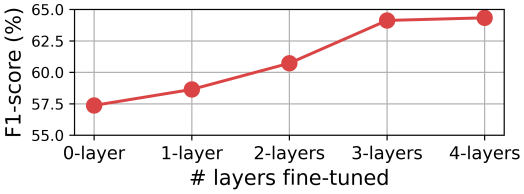


Fig. 10: Partial fine-tuning results. "0-layer" (i.e., Linear Probing) denotes just training the classifier while freezing the encoder weights. "4-layers" denotes full fine-tuning.

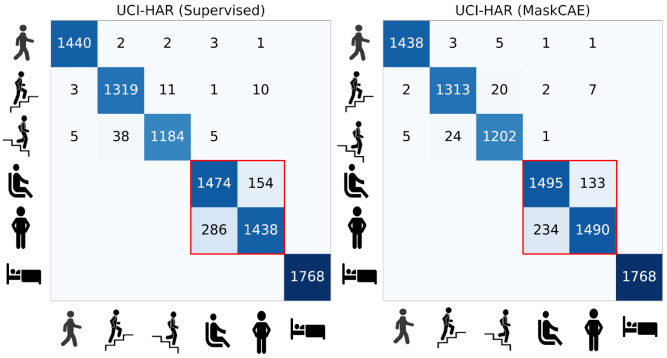


Fig. 11: Confusion Matrices. The vertical axis represents the true label, while the horizontal axis represents predicted label.

D. Confusion matrix

To clearly show MaskCAE's potential in recognizing confused activities, we further calculate the confusion matrices

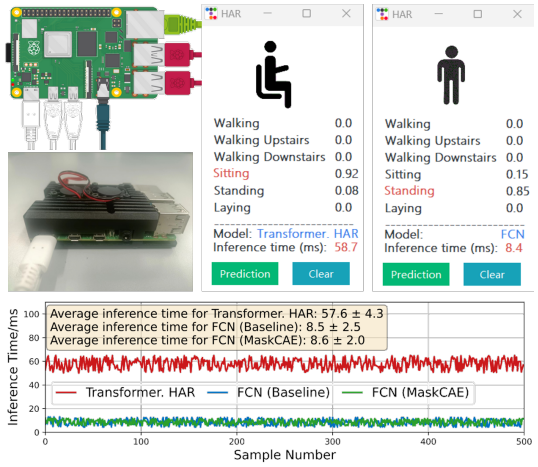


Fig. 12: HAR Model deployment on a Raspberry Pi platform.

for six types of activities selected from UCI-HAR set (also see details in Section. IV-A). Fig. 11 illustrates a pronounced misclassification between "Sitting" and "Standing", highlighted by red rectangles. From the fully supervised FCN (left panel), it is evident that the model is not easy to distinguish these two activity categories due to their similar waveforms when one subject maintains a static status, resulting in a significant number of misclassifications. In contrast, benefitting from masked sensor data modeling, the FCN trained by MaskCAE (right panel) can reduce the number of misclassifications to a much lower level.

E. Model Deployment

Since HAR has been predominantly deployed on mobile devices, the inference time is indeed crucial, beyond the aforementioned indirect metrics such as FLOPs. To provide a direct estimation for inference time, we take into account practical runtime of activity inference on a resource-constrained mobile device. Without loss of generality, an actual implementation is conducted on an embedded Raspberry Pi 4 equipped with a Quad-Core Cortex-A72 (ARM v8) 64-bit SoC running at 1.5 GHz and 4 GB LPDDR4 SDRAM, which can well support PyTorch deep learning library. Fig. 12 (top panel) presents the main user interface, including predicted softmax probabilities, the final recognized activity samples, as well as practical inference time. As depicted in Fig. 12 (bottom panel), our MaskCAE exhibits a significant advantage in edge devices, resulting in much faster inference speed (i.e., around 7× speedup) compared to transformer-based model. Because MaskCAE discards the Decoder and Feature Projections during fine-tuning, it can almost maintain the same inference speed as an end-to-end trained FCN (Baseline) without incurring extra computational burden. Thus, our MaskCAE model is more lightweight while enhancing feature representation without accuracy loss, which can align well with the practical requirements in real-world HAR situations. In addition to previous observations, it is evidently concluded that MaskCAE can provide an ideal self-supervised solution that achieves a better accuracy-cost tradeoff for on-device activity inference.

VI. Conclusion

In this paper, we introduce a novel approach by leveraging a fully convolutional model (FCN) widely employed in HAR situations to reconstruct masked sensor data. The proposed Masked Convolutional AutoEncoder (MaskCAE) provides a simple, efficient, and computationally friendly self-supervised solution for activity recognition, which is primarily composed of a sparse encoder and an information fusion decoder of hierarchical design. In the signal-wise encoder, we use sparse convolution to mask out sensor information, while only applying the reconstruction loss to these masked parts. In particular, the hierarchical decoder can effectively fuse temporal context information among different sensor channels, leading to a clear performance gain. Extensive experiments on several mainstream HAR benchmarks demonstrate that the proposed MaskCAE can consistently and significantly improve performance in self-supervised, fully supervised, and semi-supervised settings. Detailed ablation studies showcase the effectiveness of each independent component. Visual analyses illustrate the significant potential of MaskCAE in modeling temporal activity semantics. Hardware deployment indicates that our MaskCAE built on FCN can strike a better trade-off between inference time and accuracy.

In summary, our approach provides a valuable alternative to relieve the reliance on large-scale labeled datasets, since annotating sensor data is very challenging. We hope that our exploration and discovery may inspire more works in the ubiquitous HAR community to explore the potential of masked sensor reconstruction and better embrace the pretrain-finetune paradigm. For example, existing self-supervised HAR works do not consider missing device problem in multi-device HAR scenarios. One usually conducts pre-training with the whole multi-device data, e.g., including smart jacket (arm and chest), smart glasses (head), smartphone (wrist), and smart shoes (feet). However, a user might wear different devices in different scenarios: taking off his/her smart shoes when at home, or taking off his/her smart jacket if the weather is hot. This situation is realistic, since it is impossible for a user to always carry all devices in daily life. As a consequence, only part of wearable devices can be available, which can be viewed as an arbitrary subset of total devices. In this case, raw sensor data from certain unavailable devices can be completely discarded via masking to simulate missing device scenarios. Future work might extend masked sensor reconstruction to such versatile downstream tasks.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern recognition letters*, vol. 119, pp. 3–11, 2019.

[2] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[3] H. Haresamudram, A. Beedu, V. Agrawal, P. L. Grady, I. Essa, J. Hoffman, and T. Plötz, "Masked reconstruction based self-supervision for human activity recognition," in *Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 45–49.

[4] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Contrastive self-supervised learning for sensor-based human activity recognition," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2021, pp. 1–8.

[5] A. Saeed, T. Ozcelebi, and J. Lukkien, "Multi-task self-supervised learning for human activity detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, pp. 1–30, 2019.

[6] T. Plötz, "If only we had more data!: Sensor-based human activity recognition in challenging scenarios," in *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2023, pp. 565–570.

[7] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 1, pp. 857–876, 2021.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[9] H. Qian, T. Tian, and C. Miao, "What makes good contrastive learning on small-scale wearable-based tasks?" in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 3761–3771.

[10] C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo, "Exploring contrastive learning in human activity recognition for healthcare," *arXiv preprint arXiv:2011.11542*, 2020.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[12] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[14] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[15] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, "Sparse convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 806–814.

[16] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3075–3084.

[17] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *IJCAI*. AAAI Press, 2015, pp. 3995–4001.

[18] E. Kim, "Interpretable and accurate convolutional neural networks for human activity recognition," *IEEE Trans. Ind. Informatics*, vol. 16, no. 11, pp. 7190–7198, 2020.

[19] M. Zhang and A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM conference on ubiquitous computing*, 2012, pp. 1036–1043.

[20] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in *Proceedings of the international conference on internet of things design and implementation*, 2019, pp. 49–58.

[21] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Ambient Assisted Living and Home Care: 4th International Workshop, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings 4*. Springer, 2012, pp. 216–223.

[22] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. F. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *WWW*. ACM, 2017, pp. 351–360.

[23] V. S. Murahari and T. Plötz, "On attention models for human activity recognition," in *UbiComp*. ACM, 2018, pp. 100–103.

[24] S. Mahmud, M. T. H. Tonmoy, K. K. Bhaumik, A. K. M. M. Rahman, M. A. Amin, M. Shoyaib, M. A. H. Khan, and A. A. Ali, "Human activity recognition from wearable sensor data using self-attention," in *ECAI*, ser. Frontiers in Artificial Intelligence and Applications, vol. 325. IOS Press, 2020, pp. 1332–1339.

[25] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *IJCAI*. IJCAI/AAAI Press, 2016, pp. 1533–1540.

[26] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[27] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *ACM Multimedia*. ACM, 2015, pp. 1307–1310.

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[29] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

[30] H. Haresamudram, I. Essa, and T. Plötz, "Contrastive predictive coding for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–26, 2021.

[31] J. Wang, T. Zhu, L. Chen, H. Ning, and Y. Wan, "Negative selection by clustering for contrastive learning in human activity recognition," *IEEE Internet of Things Journal*, 2023.

[32] H. Haresamudram, D. V. Anderson, and T. Plötz, "On the role of features in human activity recognition," in *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 78–88.

[33] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.

[34] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan, "Designing BERT for convolutional networks: Sparse and hierarchical masked modeling," in *ICLR*. OpenReview.net, 2023.

[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[36] C. Zhang, C. Zhang, J. Song, J. S. K. Yi, K. Zhang, and I. S. Kweon, "A survey on masked autoencoder for self-supervised learning in vision and beyond," *arXiv preprint arXiv:2208.00173*, 2022.

[37] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: BERT pre-training of image transformers," in *ICLR*. OpenReview.net, 2022.

[38] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *IJCNN*. IEEE, 2016, pp. 381–388.

[39] K. Chen, L. Yao, D. Zhang, B. Guo, and Z. Yu, "Multi-agent attentional activity recognition," in *IJCAI*. ijcai.org, 2019, pp. 1344–1350.

[40] C. A. Ronao and S. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[42] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," *arXiv preprint arXiv:1904.00962*, 2019.

[43] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[44] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," *Advances in neural information processing systems*, vol. 32, 2019.

[45] L. Minhao, A. Zeng, L. Qiuxia, R. Gao, M. Li, J. Qin, and Q. Xu, "T-wavenet: A tree-structured wavelet neural network for time series signal analysis," in *International Conference on Learning Representations*, 2021.

[46] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Deep triplet networks with attention for sensor-based human activity recognition," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[47] Y. Zhang, L. Wang, H. Chen, A. Tian, S. Zhou, and Y. Guo, "If-convtransformer: A framework for human activity recognition using imu fusion and convtransformer," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–26, 2022.