

# Conditional-UNet: A Condition-aware Deep Model for Coherent Human Activity Recognition From Wearables

Liming Zhang  
George Mason University  
Fairfax, Virginia 22030, USA  
Email: lzhang22@gmu.edu

Wenbin Zhang  
University of Maryland, Baltimore County  
MD 21250, USA  
Email: wenbinzhang@umbc.edu

Nathalie Japkowicz  
American University  
Washington, DC 20016, USA  
Email: japkowic@american.edu

**Abstract**—Recognizing human activities from multi-channel time series data collected from wearable sensors has become an important practical application of machine learning. A serious challenge comes from the presence of coherent activities or body movements, such as movements of the head while walking or sitting, since signals representing these movements are mixed and interfere with each other. Basic multi-label classification typically assumes independence within the multiple activities. This is oversimplified and reduces modeling power even when using state-of-the-art deep learning methods. In this paper, we investigate this new problem, which we name “Coherent Human Activity Recognition (Co-HAR)”, that keeps complete conditional dependency between the multiple labels. Additionally, we treat Co-HAR as a dense labelling problem that classifies each sample on a time step with multiple coherent labels to provide high-fidelity and duration-sensitive support to high-precision applications. To explicitly model conditional dependency, a novel condition-aware deep architecture “Conditional-UNet” is developed to allow for multiple dense labeling for Co-HAR. We also contribute a first-of-its-kind Co-HAR dataset for head gesture recognition associated with a user’s activity, walking or sitting, to the research community. Extensive experiments on this dataset show that our model outperforms state-of-the-art deep learning methods and achieves up to 92% accuracy on context-based head gesture classification.

## I. INTRODUCTION

With the rapid development and lower cost of wearable devices with embedded sensors, a plethora of new applications, such as healthcare [1], authentication [2], robotic control [3], virtual/augmented reality [4], [5], [6] and e-learning [7] are emerging. However, a number of unique challenges need to be addressed in order to best harness the promise of such applications and enable the widespread use of wearable technology. In theory, any number of devices that detect posture could be used and mounted to any position on a human body, a practice that would result in a very large search space. In addition, people prefer to have limited devices with multiple functions, such as smart phone, virtual reality headset, smart glasses, or wireless headphones, instead of wearing multiple devices at the same time. Another challenge is that the body moves simultaneously during daily activities and generate complicated mixed signals for the limited devices mounted on the body. These multiple human activities and movements interfere with each other

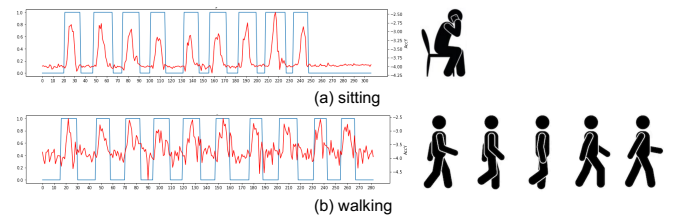


Fig. 1: A toy example of coherent human activity recognition and signals (blue lines are boolean head gesture labels, red lines are one accelerometer data): a) performing a gesture under sitting; b) performing a gesture under walking.

interactively. For example, it is important to recognize head gestures during walking (e.g. in Figure 1) using embedded sensors at only one location of a Virtual Reality headset. The upper left red line of representing a denoised accelerometer signal during sitting in Figure 1 shows a clear pattern, while the lower left red line representing the same accelerometer signal during walking is less clear. As we can see, other body parts generate stronger inertia than the simultaneous head movements while the user is walking. Previous works on head gestures [3], [8], [5] only focus on a controlled experiment environment in which users only sit still, or simply do not consider such coherent activities at all. In this work, this kind of challenging tasks with coherent interfering movements, formally defined later as “Coherent Human Activity Recognition (Co-HAR)”, will be comprehensively studied and specially-addressed.

Recent developments in deep learning shed a light on human activity recognition research, since deep learning allows to learn latent features using deep structures such as convolution layers, pooling layers and embedding layers [9]. It usually requires much less or even no effort on feature extraction than the models that pre-date deep learning. In an end-to-end fashion, deep learning models have better generality which perform well for different data without domain-specific work and result in shorter development cycles. However, deep learning for the Co-HAR problem has been highly under-explored.

Beyond naively transferring deep methods to Co-HAR, some critical technical challenges prevent current deep archi-

textures from obtaining a better generalization, including the fact that: 1) *the single location of sensors has mutual impact on signals*. As discussed, sensors mayb be placed only in a headphone over a user’s head. It is impractical to ask a user to wear sensors all over their body in real-world scenarios. In addition, it is also technical difficult to exactly separate signals and reduce mutual impacts for existing basic multi-label classification [10], [11]; 2) *the imbalanced domination of different activities could fade away the signals of the other activities*. Sensors could have different sensitive levels to different body movements and the dominating movement might not be the one under investigation in the case of study. For example, in the head gesture problem (in Figure 1) with sensors placed in a headphone, walking generates stronger signals in the forward inertia than head movements. However, head gestures are more critical for most applications like Virtual Reality. The current models might have limited power in such scenarios. 3) *the multi-label window problem for activities of various duration*. The time steps in one window may not always share the same ground truth label, and the duration of an activity always varies in different windows. Mixing of ground truth labels not only creates difficulty for underlying models but also reduces the flexibility of usages due to a whole set of hyper-parameters to be considered, such as the best window length, sampling stride and window labelling strategy.

To tackle these challenges simultaneously, we proposed a novel condition-aware deep structure, called “Conditional-UNet”<sup>1</sup>, which takes multi-channel sensor data embedded at only one location of the human body as the input and explicitly captures conditional dependence within coherent labels. The proposed framework includes a novel encoding module to model the conditional dependence which could reduce the mutual impact of coherent body movement and guide the model to better learn patterns of activities with imbalanced domination. Since it follows the dense labeling approach [12], it not only avoids the multi-label window problem, but also aims to classify multiple dense labels which is more challenging than previous single dense labeling works. The major contributions of our work are summarized as follows:

- We address a more challenging problem, so-called “Coherent Human Activity Recognition (Co-HAR)”, which classifies coherent activity labels with complete conditional dependency assumption comparing instead of simple independent multi-label assumption, and uses multi-channel time-series data from one wearable device at only one location of the human body.
- A novel condition-aware deep classification model, called “Conditional-UNet”, is developed to better densely classify coherent labels. Conditional-UNet explicitly models conditional dependence through a novel deep structure, including a new encoding module with specially-designed gradient-permitted sampling and embedding operations and a UNet-based decoding module.
- The contribution of a new dataset for the Co-HAR prob-

lem. To conduct experiments, we build an Arduino-based device to collect data and label head gestures and walk/sit condition.

- Extensive experiments show that our proposed Conditional-UNet outperforms existing state-of-the-art UNet model, and achieves up to 92.06% of accuracy and 87.83% of F1 score over head gesture classification.

## II. RELATED WORK

**Feature extraction based methods.** Models that pre-date deep learning rely heavily on hand-crafted features (e.g., mean, variance, kurtosis, or other kinds of indexes) [8], [2], motion (e.g., physical laws) [4] and transform-based feature (e.g., wavelet [13], fourier transform [14]). Exacted features are then fed to classifiers such as Support Vector Machines [4], Boosting Tree [8] and Hidden Markov Model [4]. These approaches usually work well for a specific type of tasks and fails for other types of applications.

**Deep learning based dense labeling.** With the advancement of deep learning methods, the applications of deep learning to HAR using data from wearable sensors are relatively new. More and more works propose to utilize some kinds of deep learning methods [12], [15], [16]. The success of deep-learning-based methods comes from their high expressiveness in learning underlying complex principles directly from the data in end-to-end fashion without handcrafted rules. Another most recent advancement in using deep learning is Dense Labeling [12] which uses a fully convolutional network [17] to label each sample instead of a sliding window. It avoids the segmentation problem in most of conventional methods. Another work [18] achieves the same goal of dense labeling but utilizes another deeper structure called “UNet”. However, these works still assumes a single activity label rather than coherent activities.

**Multi-label classification.** A recent study [1] tried to classify multiple overlapping activity labels for each slicing window using deep neural networks, but predicted labels do not have explicitly consideration of conditional dependence of different activities in that work. Another recent work [11] converts multiple labels into one label with all classes from different labels to solve the multi-label classification. More works exists, and we point readers to more details in other survey papers in this area [19], [20]. However, there is no existing works considering coherent multiple labels which model conditional dependency within them.

In summary, deep learning methods, including the state-of-the-art UNet, are actively researched in exiting works with better performances than conventional methods. However, to the best of our knowledge, there is no work considering conditional dependency in multiple dense labels beyond simple multi-label classification. Next, we would formally define our problem.

## III. PROBLEM DEFINITION

A set of sequences  $\mathcal{D} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_i^N, \forall \mathbf{X}^{(i)} \in \mathbb{R}^{K \times T^{(i)}}, \mathbf{Y}^{(i)} \in \mathbb{R}^{H \times T^{(i)}}$ , which contains a multivariate

<sup>1</sup>published in <https://github.com/tongjiyiming/Conditional-UNet>

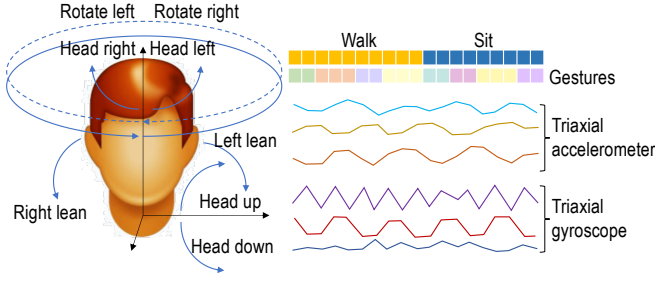


Fig. 2: A example task of Co-HAR: recognizing head gesture using accelerometer and gyroscope sensors on headphone. It has two coherent labels: head gesture label and walk/sit label.

sequence  $\mathbf{X}^{(i)}$  which have  $K$  variables of sensors and the time length of each sequence is  $T^{(i)}$ . Here, each time step  $t \in \{1, \dots, T^{(i)}\}$  is normally referred as a sample. Correspondingly,  $\mathbf{Y}^{(i)}$  is a multi-label sequence with  $H$  labels. For each label  $h \in \{1, \dots, H\}$ , there are  $C_h$  different numbers of classes for this label  $h$ , so an element  $\mathbf{Y}_{h,t}^{(i)} \in \{1, \dots, C_h\}$ , where  $\mathbf{Y}_{h,t}^{(i)} = 1$  usually define for a null label (e.g. no hand gesture is performed). The sequence index  $(i)$  is dropped in later parts whenever it is clear that we are referring to terms associated with a single sequence. We define our problem as follow:

**Definition 1 (Coherent Human Activity Recognition):** is a multi-label classification problem with conditional dependency assumption within joint multiple coherent labels and has a goal to minimize the difference between a classifier's predicted  $H$ -label sequence  $\hat{\mathbf{Y}}^{(i)}$  for  $K$ -channel sample sequence  $\mathbf{X}^{(i)}$  and the ground truth label  $\mathbf{Y}^{(i)}$  given multi-channel time-series sequences set  $\mathcal{D}$ .

For example, in our head gesture task, a sample  $\mathbf{X}_t^{(i)}$  contain  $K = 6$  variables of sensors including tri-axial acceleration and tri-axial gyroscopes. If a sampling rate is 1 Hz, then there are  $T^{(i)} = 60$  total measures for a one-second window. Since we are interested in two types of labels, head gesture and walk/sit condition, so there are two label types  $L = 2$ . For walk label  $y_1 \in \{1, 2\}$ ,  $y_1 = 1$  if a human subject is sitting, and  $y_0 = 2$  indicates a walk activity. Similarly, for head gesture label  $y_2$ ,  $y_2 = \{1, \dots, 9\}$ , and  $y_2 = 1$  means no head gestures, and other numbers indicates other 8 head gestures, and  $C_1 = 9$ .

A key component of Co-HAR is that we model the complete joint probability of all labels  $p(\mathbf{Y}_1^{(i)}, \dots, \mathbf{Y}_H^{(i)})$  with conditional dependency, rather than simplified joint probability  $p(\mathbf{Y}_1^{(i)}) \times \dots \times p(\mathbf{Y}_H^{(i)})$  that assumed independence of all labels in the existing multi-label classification studies [10], [11]. More details are in following section.

#### IV. METHODOLOGY

In this section, the condition-aware framework of Co-HAR and detailed modeling components of Conditional-UNet are introduced. Section IV-A shows the condition-aware deep framework for Co-HAR problem and the formalization of its decomposed conditional losses. The three key components of handle joint label conditions to capture the conditional

dependence within coherent activity labels are introduced in Sections IV-B, IV-C, and IV-D respectively.

##### A. Condition-aware deep framework

In this part, condition-aware deep framework is developed firstly to build up a probabilistic understanding of Co-HAR problem, and to formalize loss with an independent loss and a series of dependent losses. In general, the goal of Co-HAR in previous Co-HAR definition 1 is to learn a joint probability of multiple labels given multi-channel sensor data, noted as  $p(\mathbf{Y}_1, \dots, \mathbf{Y}_H | \mathbf{X})$ . By an axiom of probability, joint probability of observing a sequence can be decomposed to a series of independence and dependence components in Equation 1:

$$p(\mathbf{Y}_1, \dots, \mathbf{Y}_H | \mathbf{X}) = p_{\theta_1}(\mathbf{Y}_1 | \mathbf{X}) p_{\theta_2}(\mathbf{Y}_2 | \mathbf{Y}_1, \mathbf{X}) \dots p_{\theta_H}(\mathbf{Y}_H | \mathbf{Y}_{H-1}, \dots, \mathbf{Y}_1, \mathbf{X}) \quad (1)$$

where  $\theta_i$  are parameters of each probability function  $p(\cdot)$ . This is the complete conditional relationship for Co-HAR. By assuming conditional independence between all the labels, we can simplify it to be  $p(\mathbf{Y}_1, \dots, \mathbf{Y}_H | \mathbf{X}) = p_{\theta_1}(\mathbf{Y}_1 | \mathbf{X}) p_{\theta_2}(\mathbf{Y}_2 | \mathbf{X}) \dots p_{\theta_H}(\mathbf{Y}_H | \mathbf{X})$ , which is a normal multi-label classification framework in many current works [11], [10]. However, in this work, we want to keep the conditional dependence since conditional independence assumption drop a lot of useful information. Next, the condition-aware loss function is introduced with its different components.

**Condition-aware multi-label dense classification loss:** following the common approach of Maximum Likelihood Estimation (MLE) [21] and similar to dense labeling [12], we get loss function by factorizing joint probability in Equation 1 to each temporal sample with each label, and get its logarithmic transformation as follows:

$$\mathcal{L} = \log(p(\mathbf{Y}_1, \dots, \mathbf{Y}_H | \mathbf{X})) = \sum_t^T (\log(p_{\theta_1}(\mathbf{Y}_{1,t} | \mathbf{X})) + \dots + \log(p_{\theta_H}(\mathbf{Y}_{H,t} | \mathbf{Y}_{H-1,t}, \dots, \mathbf{Y}_{1,t}, \mathbf{X}))) \quad (2)$$

where  $\log(p_{\theta_1}(\mathbf{Y}_{1,t} | \mathbf{X}))$  is log-likelihood to observe different classes of label 1 on time step  $t$ th sample. Furthermore, this log-likelihood with each class  $m$  of a label is formulated as  $\sum_m^{C_1} y_{1,t}^m \log(p_{\theta_1}(\mathbf{Y}_{1,t} = m | \mathbf{X}))$ , where  $y_{1,t}^m$  is the observed frequency of class  $m$  in all samples, and  $p_{\theta_1}(\mathbf{Y}_{1,t} = m | \mathbf{X})$  is the estimated likelihood of class  $m$  got from deep model. The calculation of estimated likelihood is done by a deep model that computes likelihood  $\hat{\mathbf{y}}_1$  using sensor data  $\mathbf{X}$ , noted as  $\hat{\mathbf{y}}_1 = f_{\theta_1}(\mathbf{X})$ , where  $\hat{\mathbf{y}}_1$  is the estimated logit vector with  $m$ th element on  $t$ th sample  $\hat{y}_{1,t}^m$  as estimated probability of multi-label categorical distribution of label 1.  $f_{\theta_1}$  is a normal MLE with more details in [21]. The difference of our condition-aware model starts from label 2. Instead of only taking  $\mathbf{X}$  as inputs for  $\mathbf{Y}_2$ , our deep model takes conditional signals of  $\mathbf{Y}_1$  as input too, noted as  $\hat{\mathbf{y}}_2 = f_{\theta_2}(\mathbf{X}, \mathbf{Y}_1)$ , or  $\hat{\mathbf{y}}_i = f_{\theta_i}(\mathbf{X}, \mathbf{Y}_1, \dots, \mathbf{Y}_{i-1})$ . It means that our condition-aware deep model should decode all previous labels as joint conditional dependence for the next label estimation. Before we show

how conditional dependence is estimated, we summarize our condition-aware loss as follows:

$$\mathcal{L} = -\frac{1}{N} \left( \sum_t^T \sum_m^{C_1} y_{1,t}^m \log(\hat{y}_{1,t}^m) + \sum_t^T \sum_m^{C_2} y_{2,t}^m \log(\hat{y}_{2,t}^m) + \dots + \sum_t^T \sum_m^{C_H} y_{H,t}^m \log(\hat{y}_{H,t}^m) \right) \quad (3)$$

$$\begin{aligned} \hat{y}_{1,t}^m &= f_{\theta_1}(X), \hat{y}_{2,t}^m = f_{\theta_2}(Y_1, X) \\ \hat{y}_{i,t}^m &= f_{\theta_i}(Y_1, \dots, Y_{i-1}, X), \forall 3 \leq i \leq H \end{aligned} \quad (4)$$

$$\underset{\Theta}{\text{minimize}} \mathcal{L} \quad (5)$$

where  $\Theta = \{\theta_1, \dots, \theta_H\}$  is the set of all deep model's parameters that minimize negative log-likelihood loss  $\mathcal{L}$  in Equation 3. **To model the joint label conditions within this condition-aware deep model, we create a chain of conditional deep models  $f_{\theta_i}$ , except the first  $f_{\theta_1}$ .** The model follows the procedure illustrated in Figure 3:

**1) Decoding module:** uses a deep decoding module  $f_{\theta_1}(X)$  to compute the logit vector of label 1 for each sample  $\mathbf{y}_1 = f_{\theta_1}(X)$ . Here, the deep encoding module is UNet [22]. UNet is originally designed for image segmentation following the idea of fully convolutional network [17]. It is a deep fully convolutional network, which internally contains multiple down-sampling convolutional layers and multiple up-sampling deconvolutional layers. It is recently used to boost performance in normal HAR task [18] as a more powerful alternative than basic fully convolutional network using in Dense Labeling [12], CNN [23], or SVM [18]. We utilize the same structure as [18] More details could be found in [18];

**2) Encoding module:** uses an encoding module to convert logit vector  $\mathbf{y}_1$  to a conditional signal. It includes three sub-modules in this module: a) a generating sub-module  $Generate(\cdot)$  to get a sampled class for the first label from the categorical distribution,  $\hat{Y}_1 = Generate(\mathbf{y}_1)$  (c.f. Section IV-B); b) an embedding sub-module  $Embed(\cdot)$  (c.f. Section IV-C) is used to project  $\hat{Y}_1$  to a continuous embedding space with  $Embed_{\phi_1}(\hat{Y}_1)$ , where  $\phi_1$  is the set of parameters; c) a merging sub-module  $Merge(\cdot)$  (c.f. Section IV-D) to merge both  $X$  and embedded signal as input into the next encoding module to get a logit vector  $\mathbf{y}_2$  of label  $Y_2$  as follows:

$$\mathbf{y}_2 = f_{\theta_2}(Merge(X, Embed(Generate(\mathbf{y}_1))))$$

For  $Y_3$ , the only difference is that it merges both  $g_{\phi_1}(\hat{Y}_1)$  and  $g_{\phi_2}(\hat{Y}_2)$  with  $X$ , the logit vector  $\mathbf{y}_3$  of label  $Y_3$  is computed as follows:

$$\mathbf{y}_3 = f_{\theta_3}(Merge(X, Embed(Generate(\mathbf{y}_1))), Embed(Generate(\mathbf{y}_2))))$$

We continue this chain of processes until it reaches the last conditional model for the last label  $Y_H$ . Here, all labels  $Y_i$  are one-hot vectors;

**3) Optimizing module:** uses all logit vector  $\hat{\mathbf{y}}_i$  to calculate the multi-label dense classification loss  $\mathcal{L}$ , and minimizes it through gradient back-propagation optimization techniques for deep neural network models, such as Adam [9], or Stochastic-

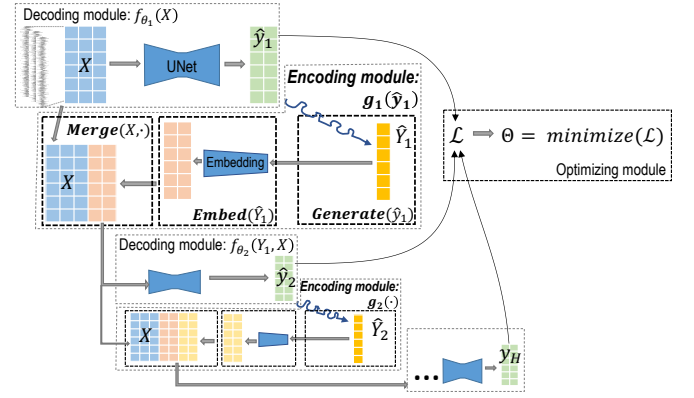


Fig. 3: Conditional-UNet: a conditional deep model with UNet module, Sampling module, and Embedding module to capture conditional dependence in coherent activities.

Gradient-Descent [9]. Since we develop our code using PyTorch [24], the Adam Optimizer is directly used to perform optimization and train our Conditional-UNet classification model.

Module 1) and 3) are conventional works with more details in other works [22], [18], [9], while Module 2) is our main structure to explicitly handle conditional dependence. We will now introduce different parts of this novel decoding module.

#### B. Gradient-permitted generating sub-module

This generating sub-module, noted as  $Generate(\cdot)$  in Figure 3, is the first step to incorporate conditional dependence information in Conditional-UNet. Its goal is to generate a sample of current label  $\hat{Y}_i$  from estimated logit  $\hat{\mathbf{y}}_i$  of which each element is a probability to get  $m$ th class, so that sampled label can be used as a conditional input for the next decoding module. The keys here are both to allow gradient back-propagation and to better process conditional dependence signal. We proposed two variants for this sub-module as follows:

**1) Naive-Max trick:** which selects the maximum probability class  $m$  in estimated logit vector  $\hat{\mathbf{y}}_i$  in Equation 6:

$$\begin{aligned} \hat{Y}_i &= \arg \max_m \hat{y}_i, \\ \forall m &\in \{1, \dots, C_h\}, \hat{\mathbf{y}}_i \in \mathbb{R}^{C_h} \end{aligned} \quad (6)$$

where  $C_h$  is the number of classes in label  $Y_i$ . This Naive-Max trick simplifies a categorical distribution to focus only on its class with maximum probability, however, it does not capture the whole distribution information. For example in Figure 4 (a), it can potentially learn a flatten distribution (differ from the true distribution in Figure 4 (c)), whose class with the max probability does not differ a lot from other classes. In this case, the distinguishing power could be vanished because of the big variance in this approach. The Naive-Max does not block gradient flow, however, it is potentially unstable because the maximum class shift a lot during training process. If a maximum-probability class is changed in a followed training iteration, the gradient flowing path changes to the other class which has maximum probability in that training iteration. This



is a huge instability disadvantage, while its advantage is its easy implementation.

2) **Gumbel-Max trick**: which implements the true process of sampling a class  $\hat{Y}_i$  from a categorical distribution using the logit vector  $\hat{y}_i$ , noted as  $\hat{Y}_i \sim \text{Cat}(\hat{y}_i)$ . This sampling process captures the full distribution information because it can still generate classes which do not have the maximum probabilities. In this way, we approximate the full categorical distribution for each class, not only the one with maximum probability. Unfortunately, this operation of sampling from categorical distribution do not have gradients, so it prevents gradients flowing in back-propagation training. The work-around solution is to use re-parameterization tricks. Specifically, we leverage a ‘‘Gumbel-Max’’ trick [25] for categorical data of labels. For example in Figure 4 (b), Gumbel-Max reduces the chance of flatten distribution like Naive-Max, while pushes the distribution to concentrate on one or a few classes and decreases the probabilities of other classes. It can get a distribution closer to the shape of true distribution (Figure 4 (c)), not only capture the peak one.

The Gumbel-Max trick is done in this way. Specifically, we create  $\mathbf{y}'_{i,t} = \tanh((\mathbf{q}^{i,t}_v + \mathbf{g})/\tau)$ , where  $\tau$  is so-called ‘‘temperature’’ hyper-parameter. Each value  $g_j$  in  $\mathbf{g}$  is an independent and identically distributed (i.i.d.) sample from standard Gumbel distribution [25].  $\mathbf{g}$  has the same dimension as  $\mathbf{y}_{i,t}$ . We generate a one-hot representation  $\mathbf{y}_{i,t}$ , whose  $j$ th element is one and all the others are zeros, where  $j$  is got as the index of the maximum element in  $\mathbf{y}'_{i,t}$ . Then, a OneHot operation is used to get the integer value of a class  $Y_{i,t} = \text{OneHot}(\mathbf{y}_{i,t})$ . In this way, gradients can be backpropagated through  $\mathbf{y}'_{i,t}$ . The same approach is also used for all the labels except the last one  $Y_H$ . The larger  $\tau$  is, the stronger uniformly regulation is imposed and the less stable gradient flows are. The typical approach is to decrease  $\tau$  as training continues and we can adopt a decrease strategy in [25] (Equations 7).

$$\begin{aligned} \mathbf{y}'_{i,t} &= \tanh((\mathbf{q}^{i,t}_v + \mathbf{g}_v)/\tau) \\ \mathbf{y}_{i,t} &= \arg \max \mathbf{y}'_{i,t} \\ Y_{i,t} &= \text{OneHot}(\mathbf{y}_{i,t}) \end{aligned} \quad (7)$$

### C. Class embedding sub-module

Generated class for each label and time  $\hat{Y}_{i,t}$  is a categorical value. Inspired in Word2Vec [26] in Natural Language Processing, we convert categorical classes to a continuous space to be processed by the following neural networks. The embedded continuous value is the conditional signal we need for conditional dependence computing of the next label  $Y_{i+1}$ . To achieve this, an embedding weight table  $\mathbf{W}_i \in \mathbb{R}^{C_h \times E_i}$  contains all learnable embedding parameters, where  $C_h$  is the number of classes in label  $Y_i$ ,  $E_i$  is a hyper-parameter of the dimension of continuous space, normally  $E_i \ll C_h$ . Here, we simply take  $E_i = \frac{C_h}{2}$ . Each label  $Y_i$  has its own embedding table  $\mathbf{W}_i$ . Embedding operation is  $\bar{\mathbf{y}}_{i,t} = \mathbf{W}_i Y_{i,t}$ , where  $\bar{\mathbf{y}}_{i,t}$  is the projected continuous vector in a continuous space.

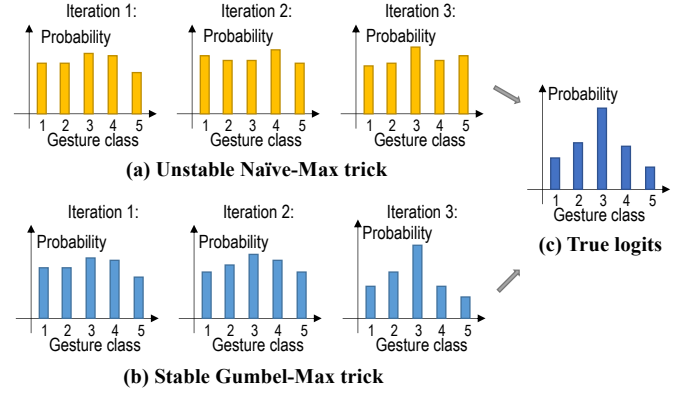


Fig. 4: Stability illustration of different Gradient-permitted generating sub-module variants: (a) three iterations of Naive-Max trick, whose peak might bounce back and forth with a flatten distribution; (b) three iterations of Gumbel-Max trick, whose learn the whole shape of distributions with more stable process; (c) the true logits of categorical distribution to be learned.

### D. Merging sub-module to capture joint conditions

The embedded vector  $\bar{\mathbf{y}}_{i,t}$  are concatenated with all previous embedded vector  $\bar{\mathbf{y}}_{1,t}, \dots, \bar{\mathbf{y}}_{i-1,t}$  and the raw sensors  $\mathbf{X}_{i,t}$ . In this way, the next label  $Y_i$ ’s joint conditions of all previous labels and sensors, noted as  $p(Y_i | \mathbf{X}, Y_1, \dots, Y_{i-1})$  in MLE before, are captured in merged vector as input for the next decoding module  $f_{\theta_i}$ . The only exception is the last label’s embedding vector  $\bar{\mathbf{y}}_{H,t}$ , which does not have concatenation operations, since we have reached the end.

In our proposed Conditional-UNet, a natural question is that what is the best order to sequentially model joint label conditions. This is just another hyper-parameter to be tuned. If there are  $H$  labels, there is potentially  $(H - 1)!$  orders. However, fortunately, there are normally not many labels in real-world applications (e.g. 2 labels in our head gesture experiment, and there are  $2! = 2$  orders to be tuned on).

## V. EXPERIMENTS

In order to demonstrate and verify the performance of the proposed Conditional-UNet for Co-HAR problem, we conduct experiments as follows: (i) collect a new dataset about head gesture under walk/sit situation through Arduino UNO and other hardwares; (ii) compare our method and its variants with state-of-the-art competing methods; (iii) a qualitative analysis to illustrate effectiveness of our proposed method.

### A. Device design and experiment settings

Hardware design: To our best knowledge, there are no dataset that is collected for Co-HAR yet, especially sensor module locates only at one location of body, so that we can only retrieve mixed signals instead of signals from multiple locations. The types of labels should be conditional dependent and interactively impact each other, so that we need to classify multiple different labels for the same sample. With this in mind, we implement an Arduino UNO module with acceleration and gyroscope sensors located on a headphone (e.g. Figure

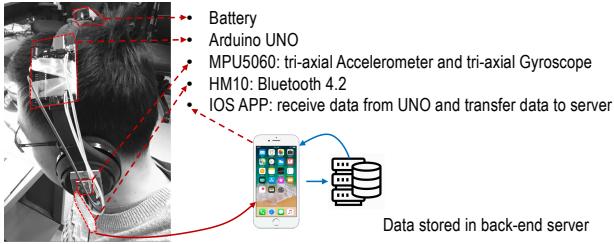


Fig. 5: Hardwares and softwares to collect conditional multiple labels and wearable sensor data of tri-axial acceleration and tri-axial gyroscopes through bluetooth communication with an IOS app and backend data storage server.

5). The data communication is done through a bluetooth HM-10 module which sends data to an IOS iphone app. Also, a camera which is not shown in the figure simultaneously records a user's head gesture and body movement video as the Arduino UNO collect sensor data. Each video is a section of either sit condition or walk condition with multiple gestures performed by a user.

To process the sensor data with a good quality, we implement the software in C++. First, it calibrates the device for the initial few seconds. Then, it starts the read of sensor data. Then, it sends the data to a registered bluetooth transmit address to allow the bluetooth module send data to an iphone through a simple IOS app that parses transmitted data and uploads to a backend server. Notice that some basic manual cleaning of the data is done aligned with the recorded videos (in 3–4 minutes duration) like clipping the starting and ending periods (about a few second duration). We then use synchronized cleaned videos (about 10 videos for each combination of head gesture classes and walk/sit classes) to manually label head gesture label and walk/sit label. In general, we collect 9 classes for head gesture label, namely left-roll, right-roll, head-right, head-left, right-lean, left-lean, head-up, head-down, and a null class of no-move (e.g. in Figure 1), and 2 classes of walk/sit label, namely if a user is walking or sitting. The baud rate is set as 9600 bits per second, and our data sample rate is chosen at  $\frac{1}{12}Hz$ . Under this setting, each ground truth head gestures contains about 20 samples in a duration of about 1.6 seconds. The summary of our collected data is in Table I. We can see that both left rolls and right rolls take much more time than other head gestures. Also, we can found that duration varies for each head gesture class. Maximum duration could be 0.3s more than the minimum duration, which is about 4 more samples. The visualization of right-roll under both sit and walk condition in Figure 6 also intuitively shows such varied duration at different times and also the strong impact from body movement under walk condition. This is an indication that Co-HAR problem is more challenging than just sit without walk.

### B. Competing methods and Conditional-UNet variants

Two competing methods including a pre-dated conventional method, and a baseline deep UNet model are used. Two variants of our proposed Conditional-UNet model are also introduced here.

TABLE I: Summary of collected head gesture data

Gesture	class number per sit section	class number per walk section	duration range of a gesture (second)
head up	9	9	1.5 - 1.8
head down	9	9	1.5 - 1.8
head left	9	10	1.5 - 1.8
head right	10	10	1.5 - 1.7
left lean	10	10	1.5 - 1.7
right lean	9	10	1.5 - 1.7
left roll	10	9	1.9 - 2.1
right roll	10	9	1.9 - 2.1
no gesture	-	-	-

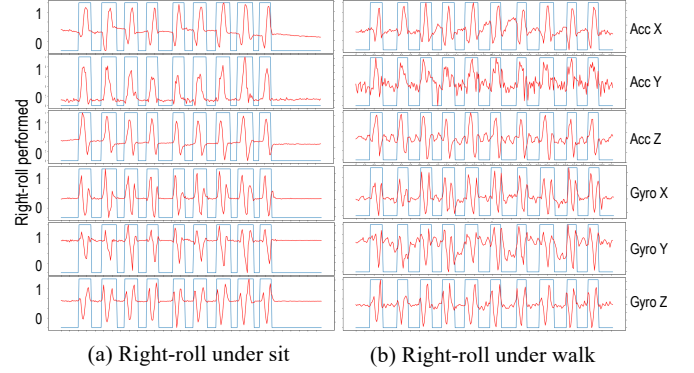


Fig. 6: Collected sensor data for right-roll gesture, red lines are sensor signals, blue line indicate if a right-roll is performed (noted 1) or not (noted 0): (a) under sit condition. The sensor signals are with clearer patterns; (b) under walk condition. The sensor signals are disturbed from significant body movements.

1) *Support Vector Machine (SVM)*: SVM is a conventional method widely used pre-dated deep learning methods. We use it as a naive baseline. The six sensors' signals are used as raw feature inputs. Two different SVM models are trained separately for head gesture label and walk/sit label.

2) *UNet baseline (UNet)*: This is a baseline multi-label classification based on UNet, a state-of-the-art deep fully convolutional network for HAR [18], which only uses one UNet decoding module to output both head gesture label and walk/sit label at the same time without any conditional dependence.

3) *Dense Head Conditioned on Dense Walk (DHcoDW)*: This model is a variant of our Conditional-UNet that first decoding modules model walk/sit label and an encoding module to encode conditional dependence of walk/sit condition, then sequentially, a second decoding module models head label. "Dense" means that both labels are classified for each sample (a.k.a. each time step).

4) *Dense Walk Conditioned on Dense Head (DWcoDH)*: This model is another variant of our Conditional-UNet that first decoding modules model head label and an encoding module to encode conditional dependence of head condition, then sequentially, a second decoding module models walk/sit label. Both labels are classified for each sample.

### C. Evaluation metric

As a classification problem, we use common accuracy score and multi-label F1 score as evaluation metric to compare

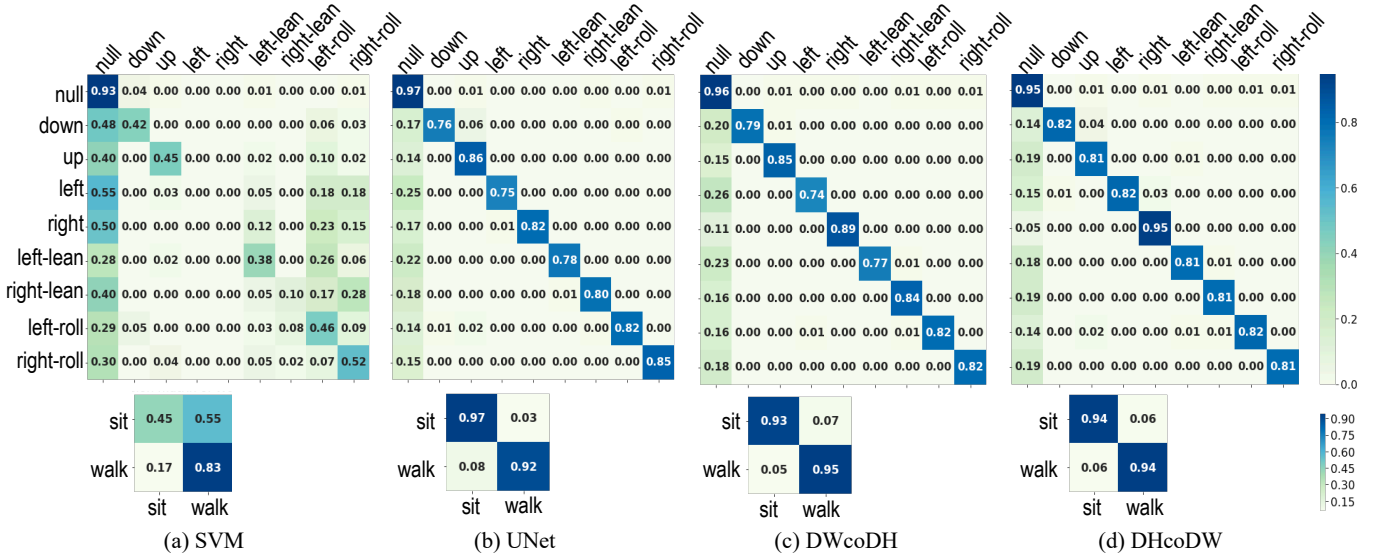


Fig. 7: Confusion matrix of multi-label classifications of different methods. Each row is normalized with total sum of each row (number of ground truth classes), and diagonal elements are true positive rate for each class.

competing method with our proposed methods. Overall, multi-label F1 score considers both precision and recall in different classes, and is better than accuracy score. We also demonstrate confusion matrix to show the performance of precision and recall for each class of a label. All the data is split into training set (about 80%) and testing set (about 20%). Accuracy and F1 scores are reported based on testing set.

#### D. Quantitative analysis

Table II contains the experiment results of accuracy and F1 scores by competing methods and variants of our proposed methods. The bolded values are the best one compared to other methods. We can see that SVM fails for head gesture label with a low F1 score of only 35.57%, while SVM performs better for walk/sit label with 66.71% F1 score. It indicates that walk/sit has much stronger signals and is easier to be classified. Basic UNet already performs very well as current state-of-the-art method with 90.46% accuracy score and 84.60% F1 score for head gesture label, and 94.94% accuracy score and 94.15% F1 score for walk/sit label. For head gesture label, DHcoDW variant perform about 2% better on accuracy (92.06%) and 3.2% better on F1 (87.83%). The performance on walk/sit label of DHcoDW is only a little worse than UNet (1.94% less on accuracy and 2.15% less on F1). For DWcoDH variant, head gesture also improve about 1% of accuracy and 2% of F1 score, and walk/sit labels are almost equally as well as UNet baseline. This results show that the DHcoDW utilizes the conditional dependence of walk/sit to better classify head gestures with a little downgrading of walk/sit label. But, DWcoDH's results show that conditional dependence of head gestures promotes less performance gain because the stronger signals of walk/sit vanish signals of head movements. This is also a good illustration of advantages of our proposed Conditional-UNet for real-world applications,

TABLE II: Model performance comparisons

Labels	Model Metric	SVM	UNet	DHcoDW	DWcoDH
Head	Accuracy	0.7516	0.9046	<b>0.9206</b>	0.9128
	F1	0.3557	0.8460	<b>0.8783</b>	0.8638
Walk/Sit	Accuracy	0.6241	<b>0.9494</b>	0.9300	0.9426
	F1	0.6671	<b>0.9415</b>	0.9201	0.9369

since head gesture label is more critical and interesting than walk/sit label in real-world applications.

The confusion matrix of different methods are shown in Figure 7, which tell more details about different methods. All confusion matrix values are normalized by total number of ground truth in this class, or in another word by the sum of each row (diagonal elements are true positive rate). The most important observation is that two variants of our proposed conditional-UNet achieve significantly gains on head gesture label, because naive UNet model mistakenly classify a large portion of head gesture as null class. Since the null class and other head gesture classes are imbalanced, the accuracy score does not quite reflect the margin of improvement as shown by confusion matrix. By comparing DWcoHD and DHcoDW variants, we can see that DHcoDW achieves a higher true positive rate except head-right, left-lean, left-roll. If we compare the walk/sit label, it is found that DHcoDW variant achieves more balance between walk class (94%) and sit class (93%), but both naive UNet and DWcoDH have low performances on walk class. This is another good demonstration that the conditional dependence design helps to get more gains by learning challenging body movements during walk conditions.

#### E. Qualitative visualization

We illustrate a few classification results here through visualization of raw sensor, ground truth, and classified classes for both head gesture label and walk/sit label in Figure 8. Each column is for each method. The first row is for head gesture label. The second row is for walk/sit label. Left Y axis



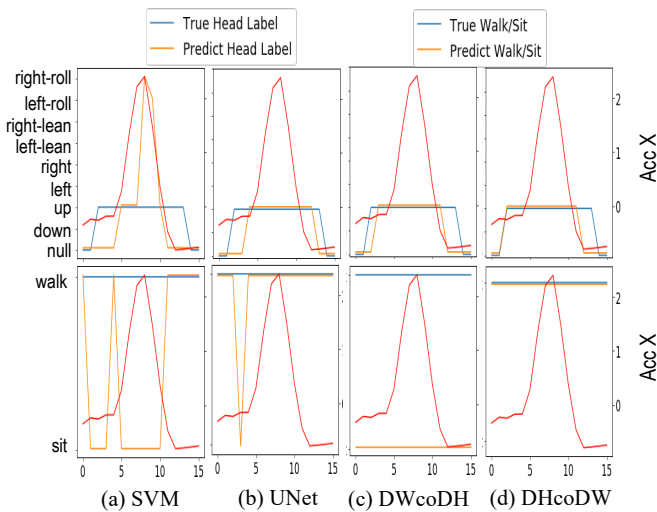


Fig. 8: Visualization of results in a selected window which show a raw sensor data (X-axis accelerometer on right Y axis) with classified and ground truth classes (on left Y axis), and X axis is the samples in this window.

of the first row plots are different classes of head gestures. Right Y axis of first and second row plots are for raw X-axis accelerometer values. The left Y axis of the second row plots are for walk/sit label. DHcoDW obtains the best result (only one sample on the right is classified wrong for head gesture label). DWcoDH unfortunately classifies wrong class for walk/sit label, while its classification for head gesture labels is very good. In practice, this error of one sample ( $\frac{1}{12}$  second) could be a minor issue for many real-world applications.

## VI. CONCLUSION

We studied a Coherent Human Activity Recognition problem, and proposed a novel condition-aware deep model “Conditional-UNet” to model the joint probability of multiple labels with explicit structures to handle conditional dependency of multiple activities in a sequential manner. The experiments we conducted show that the proposed method outperforms an older method, SVM, and a state-of-the-art UNet deep model, by 3% in F1 score. Moreover, it gets significant gains for different head gestures with a little loss in walk/sit label performance. The experiments show that our proposed Conditional-UNet successfully captures conditional dependence, as expected. In this work, a Co-HAR dataset is also contributed to the research community.

## REFERENCES

- [1] A. A. Varamin, E. Abbasnejad, Q. Shi, D. C. Ranasinghe, and H. Rezatofighi, “Deep auto-set: A deep auto-encoder-set network for activity recognition using wearables,” in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 246–253.
- [2] G. M. Parimi, P. P. Kundu, and V. V. Phoha, “Analysis of head and torso movements for authentication,” in *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. IEEE, 2018, pp. 1–8.
- [3] J. Gray, P. Jia, H. H. Hu, T. Lu, and K. Yuan, “Head gesture recognition for hands-free control of an intelligent wheelchair,” *Industrial Robot: An International Journal*, 2007.
- [4] N. N. Zolkefely, I. Ismail, S. Safei, S. N. W. Shamsuddin, and M. A. M. Arsad, “Head gesture recognition and interaction techniques in virtual reality: a review,” *International Journal of Engineering & Technology*, vol. 7, no. 4.31, pp. 437–440, 2018.
- [5] T. Hachaj and M. R. Ogiela, “Head motion classification for single-accelerometer virtual reality hardware,” in *2019 5th International Conference on Frontiers of Signal Processing (ICFSP)*. IEEE, 2019, pp. 45–49.
- [6] Y.-H. Chen and S. Itani, “Augmented reality head gesture recognition systems,” Dec. 5 2019, uS Patent App. 16/421,823.
- [7] S. P. Deshmukh, M. S. Patwardhan, and A. R. Mahajan, “Feedback based real time facial and head gesture recognition for e-learning system,” in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2018, pp. 360–363.
- [8] C.-W. Wu, H.-Z. Yang, Y.-A. Chen, B. Ensa, Y. Ren, and Y.-C. Tseng, “Applying machine learning to head gesture recognition using wearables,” in *2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2017, pp. 436–440.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [10] Y. Vaizman, N. Weibel, and G. Lanckriet, “Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–22, 2018.
- [11] R. Mohamed, “Multi-label classification for physical activity recognition from various accelerometer sensor positions,” *Journal of Information and Communication Technology*, vol. 17, no. 2, pp. 209–231, 2020.
- [12] R. Yao, G. Lin, Q. Shi, and D. C. Ranasinghe, “Efficient dense labelling of human activity sequences from wearables using fully convolutional networks,” *Pattern Recognition*, vol. 78, pp. 252–266, 2018.
- [13] W. K. Chung, X. Wu, and Y. Xu, “A realtime hand gesture recognition based on haar wavelet representation,” in *2008 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2009, pp. 336–341.
- [14] H. M. Gamal, H. Abdul-Kader, and E. A. Sallam, “Hand gesture recognition using fourier descriptors,” in *2013 8th International Conference on Computer Engineering & Systems (ICCES)*. IEEE, 2013, pp. 274–279.
- [15] A. Ignatov, “Real-time human activity recognition from accelerometer data using convolutional neural networks,” *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.
- [16] A. Ferrari, D. Micucci, M. Mobilio, and P. Napolitano, “Human activities recognition using accelerometer and gyroscope,” in *European Conference on Ambient Intelligence*. Springer, 2019, pp. 357–362.
- [17] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [18] Y. Zhang, Z. Zhang, Y. Zhang, J. Bao, Y. Zhang, and H. Deng, “Human activity recognition based on motion sensor using u-net,” *IEEE Access*, vol. 7, pp. 75 213–75 226, 2019.
- [19] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, “Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges,” *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.
- [20] 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,” *arXiv preprint arXiv:2001.07416*, 2020.
- [21] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [22] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [23] W. Xu, Y. Pang, Y. Yang, and Y. Liu, “Human activity recognition based on convolutional neural network,” in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 165–170.
- [24] N. Ketkar, “Introduction to pytorch,” in *Deep learning with python*. Springer, 2017, pp. 195–208.
- [25] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” *arXiv preprint arXiv:1611.01144*, 2016.
- [26] X. Rong, “word2vec parameter learning explained,” *arXiv preprint arXiv:1411.2738*, 2014.