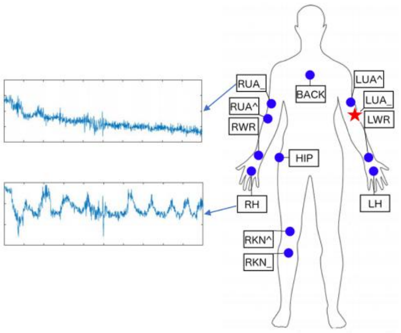
Deep Transfer Learning for Cross-domain Activity Recognition

1. What is the Problem

1.1. Problem introduction(问题引出)

There are different activity patterns of different body parts, so sensors can be put on them to collect activity data and then build machine learning models. The combination of signals from different body parts can be used to reflect meaningful knowledge such as person’s detailed health information [18] and working states [33]. Unfortunately, the real situation is that we either do not want all body parts to be equipped with sensors, or the data on certain body part may be easily missing.

如何根据其他部位佩戴的传感器的信息来预测某个没有佩戴传感器部位的信息?



如图，假定红点表示的部位（LWR）没有相应的传感器信息，而蓝点部位的信息是可获取的，如何通过蓝点部位的信息来预测出红点部位的行为?

1.2. 挑战性

1.2.1 如何挑选与 target domain 最相似的 source domains？

Firstly, we do not know which body parts are most similar to the target domain since the

sensor signals are not independent, but are highly correlated because of the shared body structures and functions. If we use all the body parts, there is likely to be negative transfer [32] because some body parts may be dissimilar.

1.2.2. target domain 没有 labels

Secondly, we only have the raw activity data on the target domain without the actual activity labels, making it harder to measure the similarities.

1.2.3. 怎么实现 transfer

Thirdly, even we already know the similar body parts to the target domain, it is also difficult to build a good machine learning model using both the source and target domains.

1.2.4. 不同人怎么挑选相似的部位

Fourthly, when it comes to different persons, there are also similar body parts across persons.

1.3. Problem definition(问题定义)

Assume we have an activity domain $\{\mathrm{D}_t^j\}_{j=1}^{n_t}$ as the target domain which we want to learn its corresponding activity labels y_t , i.e. to predict the person's activity state based on the sensor signals. Suppose we have C activity states (labels). There are M labeled source domains available: $\{\mathrm{D}_t^{\mathrm{i}}\}_{i=1}^{M}$. Each source domain $\{\mathrm{D}_t^{\mathrm{i}}\}_{i=1}^{M} = \{x_s^j, y_s^j\}_{j=1}^{n_s^i}$. Note that the data distributions are not the same, i.e. $\mathrm{P}(x_s) \neq \mathrm{P}(x_t)$. We need to design algorithms to:

1) select the best K(K < M) source domains (we denote them as $\mathrm{D}_s(K)$),

2)perform effective transfer learning from $\mathrm{D}_s(K)$ to D_t in order to

Obtain y_t .

2. Method(方法)

2.1. USSAR(Unsupervised Source Selection for Activity Recognition)

For multiple source selection, we calculate the distance between available sources and the target domains to select the most similar source domains. Our intuition is that the sensor signals may consist of generic and specific relationships about the body parts: the generic relationship means the data distance between two signals such as the Euclidean distance or cosine similarity; the specific relationship refers to the similar moving patterns or body functions between two body parts. By calculating these two significant distances, we can correctly measure the distances between different body parts, and thus select the right source domains. Our algorithm does not depend on the availability of the target domain labels. We call this algorithm Unsupervised Source Selection for Activity Recognition (USSAR).

USSAR 的大致思路就是构建一个距离函数，来表示不同 domain 之间的相似度。通过距离函数，我们就可以挑选出与 target domain 最接近的 source domains 来完成 transfer learning。

2.1.1. Distance definition(距离定义)

Our USSAR well considers the generality and specificity of activity while selecting source domains. Generality means that we have to seek the general relation between activity data, which is a common problem in machine learning. More importantly, specificity means that we should consider the specific information behind different activity domains. Formally, if we denote D(A, B) as the distance between domains A and B, then it can be represented as:

$$D(A,B) = D_g(A,B) + \lambda D_s(A,B), \qquad (1)$$

where $D_g(A,B)$ is the general distance (д for general) and $D_s(A,B)$ is the specific distance (s for specific). λ ∈ [0, 1] is the trade-off factor between two terms.

我们可以通过定义一个距离公式（如式 1），来判断 domains 之间的样本分布的相似度。

2.1.2. Generality distance $D_g(A,B)$

The general distance $D_g(A,B)$ can be easily computed by the well-established A-distance [2]:

$$D_g(A,B) = 2(1 - 2\epsilon), \qquad (2)$$

where ϵ is the error to classify domains A and B. In order to obtain ϵ, we train a linear binary classifier h on A and B, where A has the label +1 and B has the label l1 (or vice versa). Then, we apply prediction on both domains to get the error ϵ.

$D_g(A,B)$可以通过构建一个二元分类器，让分类器去对来自 domain A 和 domain B 的样本数据进行分类， 得到分类的错误率ϵ，带入式 2 得到$D_g(A,B)$。如果ϵ越小，表示分类错误率越低，说明 domain A 和 domain B 越容易区分，所以 domain A 和 domain B 的 generality distance $D_g(A,B)$越大。

2.1.3. Specificity distance $D_s(A,B)$

The specific distance $D_s(A,B)$ is composed of two important aspects from activity recognition: the semantic and the kinetic information.

Semantic distance: We basically give each source domain a weight w ∈ [0, 1] indicating its spatial relationship with the target domain. Therefore, if there are M source domains available, there will be M weights: $\{w_i\}_{i=1}^{M}$. Currently, the weighting technique is only based on human experience

$$d_s(A,B) = \mathbb{E}\left[\sum_{\mathbf{a},\mathbf{b}} \frac{w_{\mathbf{a}}\mathbf{a} \cdot w_{\mathbf{b}}\mathbf{b}}{|w_{\mathbf{a}}\mathbf{a} \cdot w_{\mathbf{b}}\mathbf{b}|}\right]. \tag{5}$$

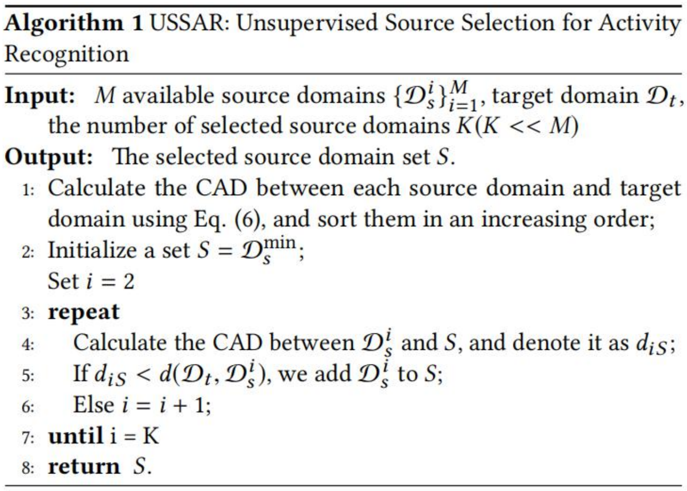
where a, b are the basic vectors in A and B, respectively. E[·] is the expectation of samples. Note that when one domain is the target domain, its weight can be set to 1.

2.1.4. USSAR algorithm

Overall distance: we combine the general distance (Eq. 2) and the specific distance (Eq. 5) to get the signal distance expression:

$$d(A,B) = 2(1-2\epsilon) + \lambda\mathbb{E}\left[\sum_{\mathbf{a},\mathbf{b}} \frac{w_{\mathbf{a}}\mathbf{a} \cdot w_{\mathbf{b}}\mathbf{b}}{|w_{\mathbf{a}}\mathbf{a} \cdot w_{\mathbf{b}}\mathbf{b}|}\right]. \tag{6}$$

Note that this distance measurement does not rely on the labels of the target domain. We call this distance the Context Activity Distance (CAD). Once we have the CAD, we can perform source selection from many available source domains.



Algorithm 1 USSAR: Unsupervised Source Selection for Activity Recognition

```
Input: $M$ available source domains $\{\mathcal{D}_s^i\}_{i=1}^M$, target domain $\mathcal{D}_t$,
    the number of selected source domains $K (K << M)$
Output: The selected source domain set $S$.
Calculate the CAD between each source domain and target
domain using Eq. (6), and sort them in an increasing order;
Initialize a set $S = \mathcal{D}_s^{\min}$;
Set $i = 2$
repeat
    Calculate the CAD between $\mathcal{D}_s^i$ and $S$, and denote it as $d_{iS}$;
    If $d_{iS} < d(\mathcal{D}_t, \mathcal{D}_s^i)$, we add $\mathcal{D}_s^i$ to $S$;
    Else $i = i + 1$;
until i = K
return $S$.
```

2.2. TNNAR(Transfer Neural Network for Activity Recognition)

2.1.1. Frame(架构)

After obtaining the right source domains via the USSAR algorithm, we propose a Transfer Neural Network for Activity Recognition (TNNAR). TNNAR is an end-to-end neural network to perform knowledge transfer across different domains. The important thing is that in order to reduce the distance between two domains, we add an adaptation layer in the network to calculate the adaptation loss, which can be optimized jointly with the classification loss.

The main structure of TNNAR is two convolutional blocks with max-pooling operations, one LSTM layer, and two fully-connected layers. We use the convolutional layers to extract the spatial features from the original activity data. The LSTM layer is mainly for capturing the time features [39]. The fully-connected layers are used for nonlinear transformation. Finally, a softmax function is applied for classification.

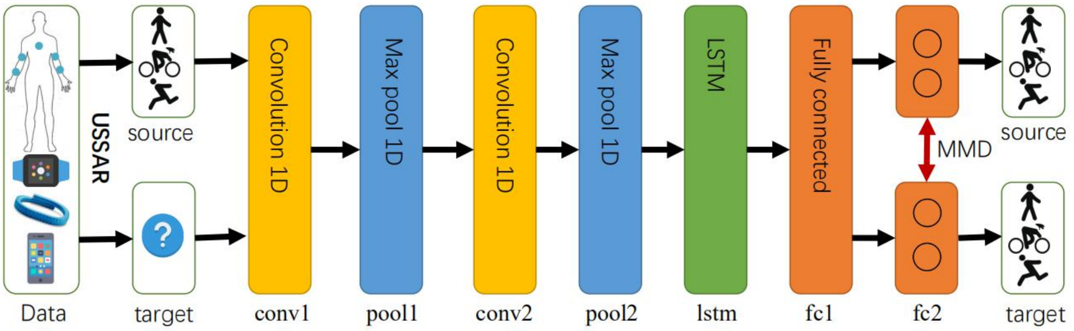


Figure 3: The structure of Transfer Neural Network for Activity Recognition (TNNAR).

这一段阐述了 TNNAR 的网络架构，卷积层 Convolution layers 对传感器信息进行特征提取 feature extraction，LSTM layer 使得 TNNAR 具有时序“记忆”，连接层

最终进行分类。

Loss function 损失函数的定义要考虑到两个方面：

1) 提高预测准确率

2) 减低不同 domains 之间样本数据分布的 discrepancy

2.1.2. Loss function

$$f(\mathbf{x}) = \ell_c(\mathbf{x}) + \mu\ell_a(\mathcal{D}_s, \mathcal{D}_t), \tag{7}$$

$$\ell_c = \min_{\Theta} \frac{1}{n_b} \sum_{i=1}^{b} J(\Theta(\mathbf{x}_i^b), y_i^b), \tag{8}$$

where J is the cross-entropy function. (x_i^b , y_i^b) denotes all the labeled samples from the source domain. As for the adaptation layer, we adopted the well-established Maximum Mean Discrepancy (MMD) [5] as the measurement to reduce the discrepancy between domains. The MMD distance between distributions p and q is defined as $d^2(p,q) = (\mathrm{E}_p[\phi(z_s)] - \mathrm{E}_q[\phi(z_t)])^2_{H_k}$ where H_K is the reproducing kernel Hilbert space (RKHS) induced by feature map ϕ(·). Here, E[·] denotes the mean of the embedded samples. Therefore, the MMD distance between the source and target domain is

$$MMD(\mathcal{D}_s, \mathcal{D}_t) = \|\mathbb{E}[\mathbf{x}_s] - \mathbb{E}[\mathbf{x}_t]\|^2_{\mathcal{H}_K}. \tag{9}$$

We train the TNNAR using mini-batch Stochastic Gradient Descent (SGD) strategy. The gradient can be calculated as

$$\Delta_{\Theta^l} = \frac{\partial J(\cdot)}{\partial \Theta^l} + \mu \frac{\partial \ell_a(\cdot)}{\partial \Theta^l}. \tag{10}$$

3. Innovation

Conventional machine learning approaches tend to solve it by subsequently performing preprocessing procedures, feature extraction, model building, and activity inference. However, they all assume that the training and test data are with the same distribution. As for CDAR where the training (source) and the test (target) data are from different feature distributions, those conventional methods are prone to under-fitting since their generalization ability will be undermined.

传统的算法无法应对 training data 和 test data 分布不一致的情况，导致结果会 under-fitting。

3.1. 第一个提出了 USSAM 同于挑选与 target domain 分布最相似的 source domains

We propose the first unsupervised source selection algorithm for activity recognition. USSAR measures both the general and specific characteristics of activity information, hence it is capable of capturing the profound relationship between different domains.

3.2. 提出了 TNNAR 用于同时提高 adaptation 和 classification

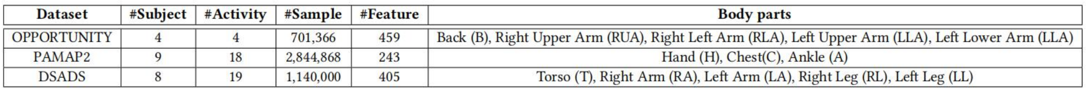
We propose an end-to-end transfer neural network for cross-domain activity recognition. Different from existing deep transfer learning methods that need to extract features from human knowledge, our TNNAR can simultaneously perform classification and adaptation between two activity domains on the original data.

4. experiment

4.1 Datasets and Setup

There are three public datasets used in [40]: OPPORTUNITY dataset (OPP) [8], PAMAP2 dataset (PAMAP2) [34], and UCI daily and sports dataset (DSADS) [1].

Table 1: Brief introduction of three public datasets for activity recognition [40]



Dataset	#Subject	#Activity	#Sample	#Feature	Body parts
OPPORTUNITY	4	4	701,366	459	Back (B), Right Upper Arm (RUA), Right Left Arm (RLA), Left Upper Arm (LLA), Left Lower Arm (LLA)
PAMAP2	9	18	2,844,868	243	Hand (H), Chest(C), Ankle (A)
DSADS	8	19	1,140,000	405	Torso (T), Right Arm (RA), Left Arm (LA), Right Leg (RL), Left Leg (LL)

OPP is composed of 4 subjects executing different levels of activities with sensors tied to more than 5 body parts. PAMAP2 is collected by 9 subjects performing 18 activities with sensors on 3 body parts. DSADS consists of 19 activities collected from 8 subjects wearing body-worn sensors on 5 body parts. Accelerometer, gyroscope, and magnetometer are all used in three datasets. The transfer scenarios are obtained according to [40]. There are three scenarios that reflect different similarities between domains:

a) similar body parts of the same person,

b) different body parts of the same person,

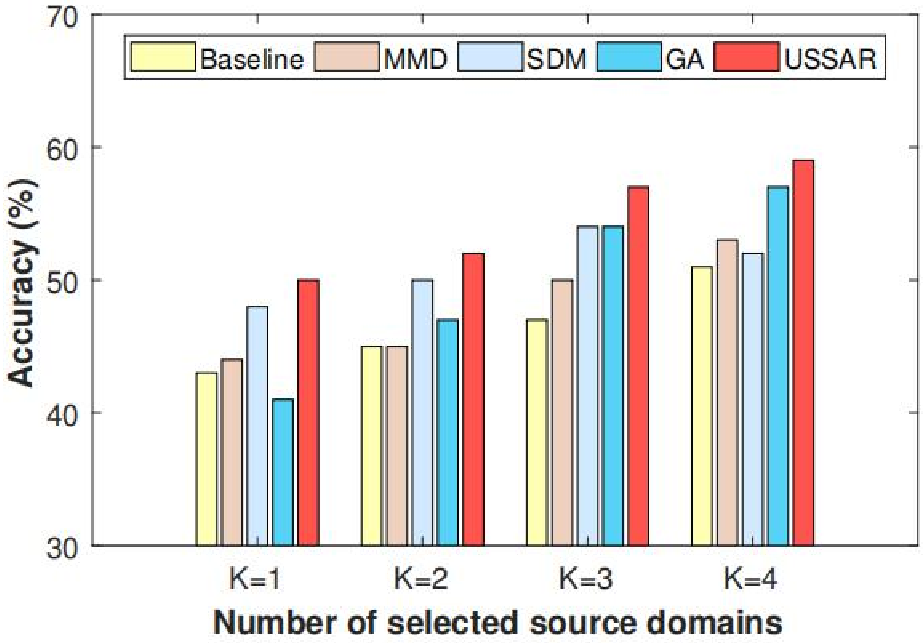
c) similar body parts of different person.

In the sequel, we use the notation A → B to denote labeling the activity of domain B using the labeled domain A. In total, we constructed 22 tasks. Note that there are different activities in three datasets. For scenario a) and b), we simply use all the classes in each dataset; for scenario c) which is cross-dataset, we extract 4 common classes for each dataset (i.e. Walking, Sitting, Lying, and Standing). In addition, we did not include the scenario ‘different body parts of different person’ since 1) all the methods perform poorly in that scenario, and 2) that scenario does not have reasonable feasibility in real applications.

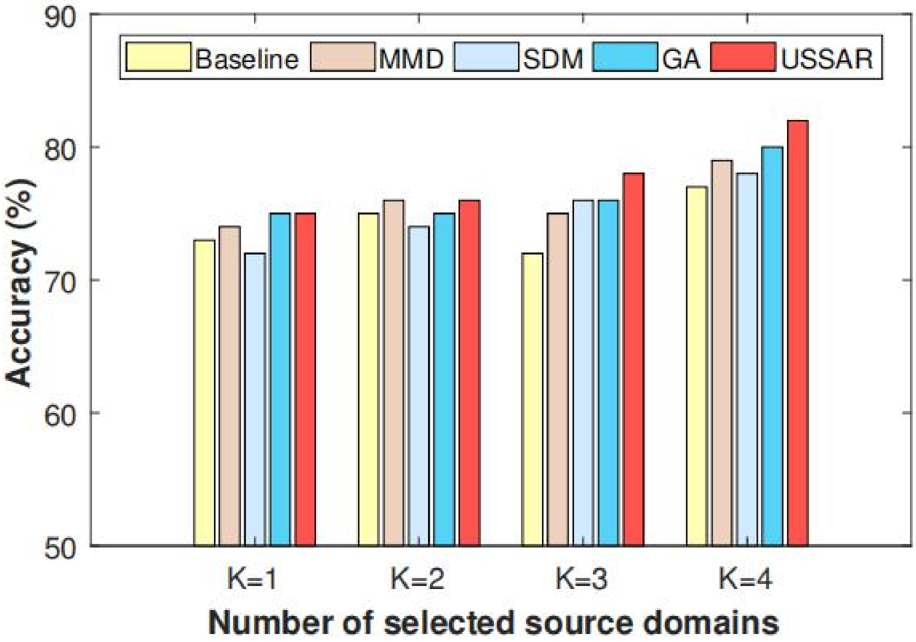
4.2 Evaluation of USSAR

We compare USSAR with several source selection techniques:

- A-distance [2], which is selected according to the top K smallest A-distances. This distance is acting as the baseline.
- SDM: subspace disagreement measurement [15], which is computing the distance between domains based on the principle angle [17].
- GR: Greedy algorithm [3], which is selecting sources using a greedy technique.
- MMD: Maximum Mean Discrepancy [5], which is a popular metric to measure the distance.



(a) Torso as the target domain



(b) Right Arm as the target domain

4.3 Evaluation of TNNAR

4.3.1. Comparison

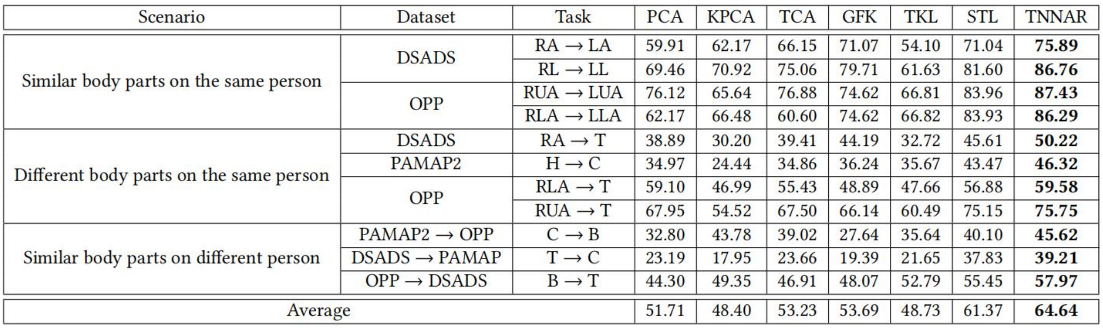
In order to test the performance of TNNAR, we conducted two experiments: transfer learning on single source domain, and transfer learning on multiple source domains. We compare TNNAR with the following methods:

- PCA: Principal component analysis [13].
- KPCA: Kernel principal component analysis [13].
- TCA: Transfer component analysis [31].
- GFK: Geodesic flow kernel [15].
- TKL: Transfer kernel learning [25].
- STL: Stratified Transfer Learning [40].

4.3.1. Implementation

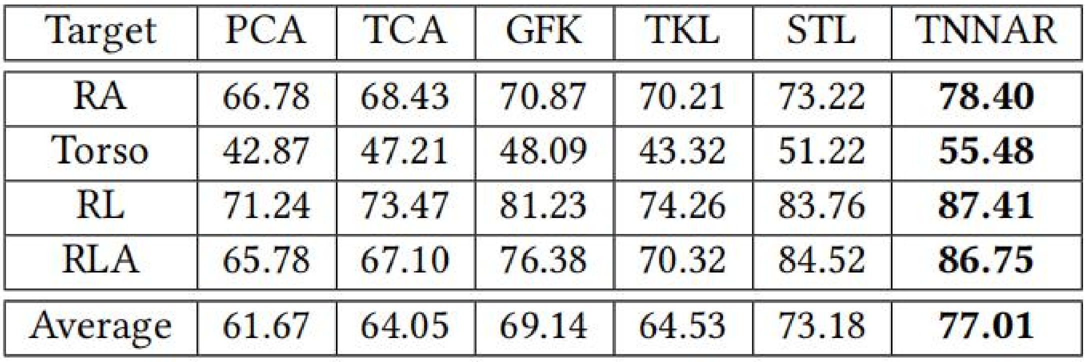
The implementations of all comparison methods are following [40]. Different from these work which exploited feature extraction according to human knowledge, we take the original signal as the input. For TNNAR network, we set the learning rate to be 0.001 with a dropout rate of 0.8 to prevent over-fitting. The batch sizes for source and target domains are 64. Note that although we selected K source domains, we basically combine them into one large source domain. Since the sensor signal is a multi-channel reading, we treat each channel as a distinct signal and perform 1D convolution on it. Totally, there are 9 channels (i.e. 3 accelerometers, 3 gyroscopes, and 3 magnetometers). The convolution kernel size is 64 × 1 with the depth 32. Other parameters of the neural network are set accordingly. For the MMD measurement, we take the linear-time MMD as in [16]. Note that in both of the two experiments, all of the comparison methods used the same source and target domains. For the single source domain situation, we follow the settings in [40] and report the results in Table 2. For the experiments on multiple source domains, we extend the results in the last section and set K = 3 to select the source domains by USSAR. The results are in Table 3.

Table 2: Classification accuracy (%) of TNNAR and other comparison methods on single source transfer tasks.



Scenario	Dataset	Task	PCA	KPCA	TCA	GFK	TKL	STL	TNNAR
Similar body parts on the same person	DSADS	RA → LA	59.91	62.17	66.15	71.07	54.10	71.04	**75.89**
		RL → LL	69.46	70.92	75.06	79.71	61.63	81.60	**86.76**
	OPP	RUA → LUA	76.12	65.64	76.88	74.62	66.81	83.96	**87.43**
		RLA → LLA	62.17	66.48	60.60	74.62	66.82	83.93	**86.29**
Different body parts on the same person	DSADS	RA → T	38.89	30.20	39.41	44.19	32.72	45.61	**50.22**
	PAMAP2	H → C	34.97	24.44	34.86	36.24	35.67	43.47	**46.32**
	OPP	RLA → T	59.10	46.99	55.43	48.89	47.66	56.88	**59.58**
		RUA → T	67.95	54.52	67.50	66.14	60.49	75.15	**75.75**
Similar body parts on different person	PAMAP2 → OPP	C → B	32.80	43.78	39.02	27.64	35.64	40.10	**45.62**
	DSADS → PAMAP	T → C	23.19	17.95	23.66	19.39	21.65	37.83	**39.21**
	OPP → DSADS	B → T	44.30	49.35	46.91	48.07	52.79	55.45	**57.97**
Average			51.71	48.40	53.23	53.69	48.73	61.37	**64.64**

Table 3: Accuracy (%) of multiple source domains



Target	PCA	TCA	GFK	TKL	STL	TNNAR
RA	66.78	68.43	70.87	70.21	73.22	**78.40**
Torso	42.87	47.21	48.09	43.32	51.22	**55.48**
RL	71.24	73.47	81.23	74.26	83.76	**87.41**
RLA	65.78	67.10	76.38	70.32	84.52	**86.75**
Average	61.67	64.05	69.14	64.53	73.18	**77.01**

4.3.3. Why TNNAR is better

The reasons are three folds:

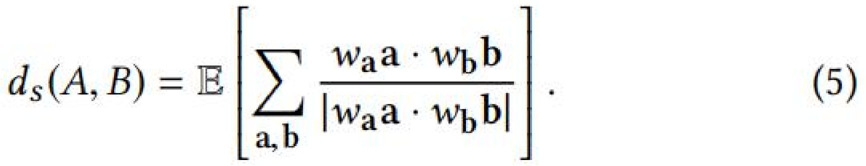
1) Other comparison methods are operated on the extracted features according to human knowledge, which may not be sufficient to capture the resourceful information of the activities. TNNAR is based on the deep neural network to automatically extract features without human knowledge. As previous work has demonstrated the effectiveness of deep neural network on feature extraction [39], it will help the network to extract more high-level features.

2) The structure of the neural network is beneficial for performing transfer learning, since the hyperparameters can be easily shared across domains.

3) The deep neural network has both the convolution and LSTM cells, which enables it to learn the spatial and time information from the activities. Therefore, the network can understand more information about the activity data.

5. 我的疑惑

5.1.



$$d_s(A, B) = \mathbb{E}\left[\sum_{\mathrm{a,b}} \frac{w_{\mathrm{a}}\mathrm{a} \cdot w_{\mathrm{b}}\mathrm{b}}{|w_{\mathrm{a}}\mathrm{a} \cdot w_{\mathrm{b}}\mathrm{b}|}\right]. \qquad (5)$$

在 2.1.3 中，公式(5)中分子和分母的 w 权重为什么不会被直接约掉？

而且绝对值符号是否有误？

5.2. 输入数据的形式是怎么样子的？

原论文提到

Note that although we selected K source domains, we basically combine them into one large source domain. Since the sensor signal is a multi-channel reading, we treat each channel as a distinct signal and perform 1D convolution on it. Totally, there are 9 channels (i.e. 3 accelerometers, 3 gyroscopes, and 3 magnetometers). The convolution kernel size is 64 × 1 with the depth 32.

比如说，假定 target domain 是右下臂，我们选取 source domains 分别是从左上臂、左上臂、左下臂提取到的传感器数据，每个部位分别记录有九个传感器数据(i.e. 3 accelerometers, 3 gyroscopes, and 3 magnetometers)。那么每个样本就是一个 dimension 为 9 的向量，为什么论文可以使用一个 size is 64 × 1 with the depth 32 的 convolution kernel 呢？输入数据组织究竟是怎么的，为什么可以用以处理图片为专长的 Convolution Network 来处理传感器数据？

5.3. MMD 的原理不清楚

这点我可能需要再了解一下相关的论文。