Multilevel Transfer Learning for Complex Work Activity Recognition in Logistic Domain

Jaime Morales
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
jaime.morales@ist.osaka-u.ac.jp

Qingxin Xia
Hong Kong University of Science and Technology (Guangzhou)
Guangdong, China
qingxinxia@hkust-gz.edu.cn

Naoya Yoshimura
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
naoya.yoshimura.work@gmail.com

Hirotomo Oshima
Corporate Manufacturing Engineering Center
Toshiba Corporation
Kanagawa, Japan
hirotomo1.oshima@toshiba.co.jp

Masamitsu Fukuda
Corporate Manufacturing Engineering Center
Toshiba Corporation
Kanagawa, Japan
masamitsu1.fukuda@toshiba.co.jp

Yasuo Namioka
Advanced Institute of Industrial Technology
Tokyo, Japan
namioka-yasuo@aiit.ac.jp

Takuya Maekawa
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
maekawa@ist.osaka-u.ac.jp

***Abstract*—Complex work activity recognition based on wearable sensors is crucial for streamlining work processes in industrial domains. Unlike basic activities such as walking or running, which involve simple repetitive motions, a complex work activity consists of discrete atomic actions such as an action of spreading a shipping label or cutting tape in a packaging task. In addition, the atomic actions sometimes involve characteristic short-term sensor data patterns. In addition, these actions can be performed in different orders by different workers to achieve similar outcomes, resulting in different long-term sensor data trends for different workers. Because multilayer networks for activity recognition may learn short-term features from shallow-level layers and long-term trends from deeper layers, we propose a new transfer learning method called multilevel knowledge transfer (MLKT), which performs level-wise source selection according to trend similarity across workers in different levels. For example, for training shallow layers, highly similar workers are selected for specific short motions (e.g., pasting a shipping label), and to train the deeper layers, workers with similar cadence are selected. This method also enables the adaptive thresholding of source data selection for each layer level during network training using the proposed adaptive level-wise discerning module.**

***Index Terms*—Activity recognition, Machine learning, Transfer learning, Logistics, Packaging task**

I. Introduction

1) Background: With the increasing trend in e-commerce, last-mile delivery and logistics centers have become key actors in the overall performance of global supply chains for large and small retailers around the world. Studies show that one in six people buys items online at least once a day, with one in four using e-shopping services at least once every two weeks [1]. It has been reported that online retail buyers have increased from 2.2 billion to 2.7 billion in the last three years alone [2]. Currently, logistics centers rely largely on manual operations performed by human employees to ensure flexible responses to the fast-paced demand changes from customers and suppliers, and this trend is expected to continue [3]–[5]. These facts indicate that streamlining processes by the employees at logistics centers can significantly improve the supply chains of many retailers worldwide.

Human activity recognition (HAR) techniques have been actively studied in the PerCom community to quantify manual activities in industrial domains, such as factories and warehouses [6], [7]. However, highly accurate models require large amounts of labeled data from a target worker, which can be expensive. In general, transfer learning is a promising approach to deal with this problem, and this study focuses on knowledge transfer for complex work activity recognition. Transfer-learning techniques for simple activity recognition have been actively studied in this community [8]–[20]. In particular, researchers have observed that the most similar source domain can be found by comparing several characteristics between the unknown target and available source domains. Then a model trained or adapted from the most similar source can be used to predict the activities performed by the target [10], [11], [16], [17]. However, applying these techniques to complex work activities is challenging.

2) Problems: Unlike simple daily activities (walking, climbing, etc.), a complex activity encompasses a sequence of smaller atomic actions. For example, to recognize the "Read label" operation in packaging tasks, a model would need to identify atomic actions, such as 'grab QR scanner,' 'scan the item label,' and 'drop the QR scanner.' In addition, the order in which these actions are usually performed must also be identified. However, previous transfer learning methods performed source selection without considering these properties. In many cases, when attempting to transfer knowledge from existing source workers to an unknown target worker,

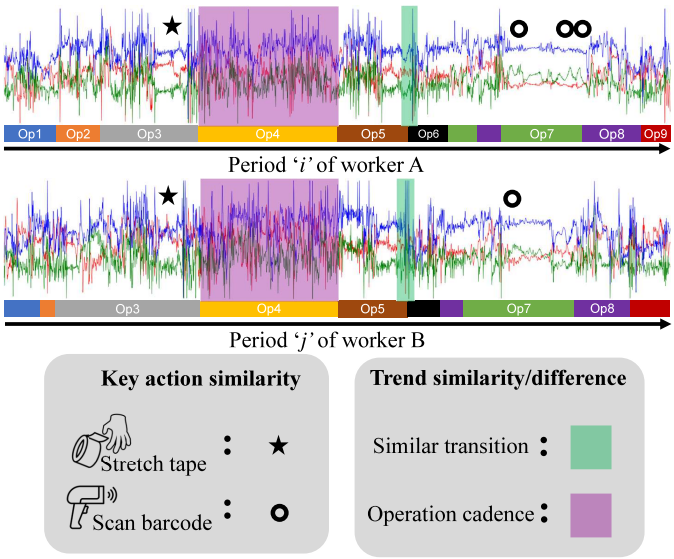

Fig. 1. Example accelerometer data from a work period for two separate workers from the LOGI dataset. Highlighted are examples of existing trend similarities and key action similarities necessary for accurate complex activity recognition.

a particular source worker who shares short-term data trends (atomic action) with a target does not always share long-term data trends (action sequences) with that target, and vice versa. For example, Figure 1 shows different workers sharing similar short-term trends (key actions and transition from Operation 5 to Operation 6), but a difference in long-term trends for some operations (e.g., Operations 4 and 7).

In addition, previous source selection methods often selected a single source for a particular target or sources exhibiting a similarity higher than an arbitrarily set threshold [11], [16], [21]. However, defining the appropriate number of sources or similarity thresholds for a particular case is difficult to determine in advance.

3) Approach: To address the two problems, we propose a new transfer learning method for complex packaging work activities with two features: (i) knowledge transfer according to the properties of the complex work, i.e., short- and long-term sensor data trends, and (ii) adaptive thresholding for source selection during network training.

As for the first feature, we focus on the differences in encoding levels present in multilevel neural networks such as convolutional neural networks (CNNs). A network learns to capture the relevant information at different levels of encoding. In general, shallow layers capture short-term trends within a work period, determining temporal locations of particular atomic actions in the sequence. In contrast, after compressing the extracted features, deeper layers capture long-term trends (e.g., action sequences). Therefore, our proposed idea is multilevel knowledge transfer where appropriate source workers are selected for 'each network level' by identifying those who share relevant characteristics based on trend similarity across workers at the network level. For example, when training the shallow layers in a network, only data from source workers with short-term data trends, i.e. atomic actions, similar to the target are employed. On the other hand, when training deep layers in the network, only data from workers whose overall features can be interpreted as having a similar cadence during packaging are used.

We propose a multilevel knowledge transfer (MLKT) technique that transfers knowledge according to the similarity between source and target workers at each encoding level. In addition, to address the drawbacks of prior transfer learning methods, we propose an Autonomous Source Discrimination Module (ASDM) that adaptively determines the similarity threshold and percentage of sources to use for adequate fine-tuning at each encoding level during network training (fine-tuning) to maximize classification performance. The ASDM determines the threshold based on the distribution of the similarity scores between the source and target workers because this distribution describes how many similar source workers are available in the training data. In the proposed method, we transfer knowledge across source and target workers by employing labeled sensor data from the source workers and unlabeled data from the target worker.

4) Contributions: We proposed a new transfer learning method for work activity recognition to address the drawbacks of prior transfer learning methods: (i) inability to consider the properties of complex works (i.e., short- and long-term trends) and (ii) inflexible source selection. The research contributions of this study are listed as follows.

- To the best of our knowledge, this is the first study to perform level-wise knowledge transfer for complex packing work activity recognition. We select appropriate source workers for each network level according to short- and long-term sensor data trends.
- We present an automated source-selection method to adapt the required number of selected sources during fine-tuning. The method is introduced to automatically define the similarity threshold and data percentage for source selection within each network level during fine-tuning.

II. Related Work

A. Human Activity Recognition in Industry

With a growing interest in smart manufacturing, many studies rely on a variety of sensors such as accelerometers to recognize and support factory activities. Xia et al. [22], [23] created a motif-based[1] particle filter to identify real-world factory activities without using labels. Maekawa et al. [24] used period motifs (corresponding to atomic actions), and an unsupervised approach to calculate the length of working periods. Tao et al. [25] applied CNNs to identify consecutive assembly operations by converting multiple-channel inertial measurement units (IMUs) and electromyographic data into spectrogram images, thereby using them to train convolutional networks. Syed et al. [26] developed a method for continuous recognition of logistics work operations using the CNN-IMU in different architectures.

[1]A motif is a characteristic sensor data segment corresponding to an atomic action.

Yoshimura et al. [27] presented LOS-Net, a lightweight CNN-based model for ordered work segmentation for assembly and packing activities. Two modules designed for complex manual works were employed: a boundary detection module and a Work Process Context Pooling (WPCP) module. The boundary detection module is trained to detect boundaries between consecutive operations. The WPCP module captures inter-activity information and intra-activity information using dilated convolution kernels. Morales et al. [6], extracted motifs corresponding to actions representative of each operation and used the them to train attention-based neural networks. Previous supervised methods require fully labeled data, whereas unsupervised methods rely on relatively consistent work operations to recognize complex operations. To the best of our knowledge, the proposed method is the first to apply knowledge transfer for complex work activity recognition in the industrial/logistic domain.

B. Transfer Learning for Human Activity Recognition

According to a literature survey for HAR, transfer learning can be categorized into three types: instance-, parameter-, and feature-based [8]. In instance-based methods, data instances from the source domains are selected based on their properties and used to train the model to be applied in the target domain. One such technique is source selection [9]–[11], [28]. Qin et al. [9] performed cross-dataset HAR transfer for inertial data collected from diverse body parts by employing an adaptive evaluation of marginal and conditional distributions to learn spatial features from source domain data, and further using incremental manifold learning to appropriately transfer knowledge from known to unknown domains by identifying the transferable temporal features. Similarly, Wang et al. [11] transferred knowledge between subjects performing sports and daily activities while wearing IMUs by performing semantic and kinetic distance calculations between the source and target subjects to identify the ideal source for knowledge transfer.

Parameter- and feature-based methods rely on transferring information from a previously trained model into a newly created model to be used in an unknown domain. Examples include domain-adversarial and domain-adaptive networks [12]–[15], [20]. For example, Khan et al. [15] proposed a heterogeneous deep convolutional neural network (HDCNN) that finds similarities across multiple domains between the source and target data to adapt the pre-trained network by minimizing the discrepancy between the two datasets after each fully connected convolutional block. For this, they used Kullback-Leibler (KL) divergence as a distance measure between the target domain feature extractor and pre-trained networks, aiming to assimilate network parameters to the new target domain. A different approach was presented by Qin et al. [20] for HAR, where a deep convolutional model is trained by fusing domain-specific and domain-invariant features for domain adaptation. Sanabria et al. [13] proposed another domain adaptation technique, namely, the contrastive generative adversarial network (Contras-GAN), for heterogeneous feature transfer and contrastive learning, to better distinguish between

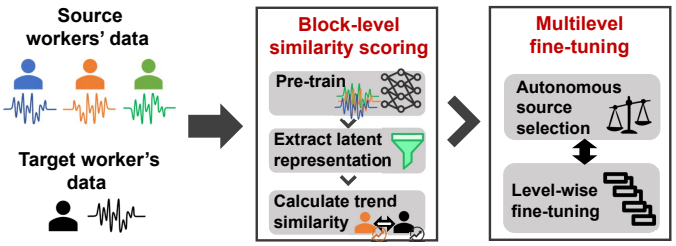

Fig. 2. Overview of Multilevel Knowledge Transfer method.

classes from multiple sources for cross-user, cross-body, and cross-sensor knowledge transfer with inertial motion data for HAR.

Recently, domain generalization approaches have gained some recognition. An example of this applied to IMU-collected data is the DAPPER method presented by Gong et al. [19]. DAPPER estimates the adaptation performance for a target domain, without requiring labeled data from the target, by training an estimator on features obtained by generating virtual source and target domain pairs and iteratively adapting them while collecting training features and classification accuracies. Meanwhile, Zhang et al. [18] presented an unsupervised approach that utilizes domain-aware representation learning to train a target model with solely unlabeled data and better generalize prediction performance across categories for image-based object recognition and classification.

While existing approaches can improve user-independent models for traditional HAR, complex action recognition (e.g., packaging operations) is difficult for these types of transfer techniques, given their lack of focus on short- and long-term trends in the complex work.

III. Multilevel Knowledge Transfer

A. Preliminaries

This method employs three-axis accelerometer data captured from workers while performing packaging work. A work period includes one iteration of the operations required to fulfill a work order, and the task is to classify each data point from the work periods corresponding to an unknown target domain. The target domain is defined as a single worker's data, comprising unlabeled test periods and only a few unlabeled reference periods J, such that J is much smaller than the number of total periods. The j-th reference period is referred to as RP_j. These periods are used as a reference to compute data similarities between the target and the source domains. Note that we do not use any labels from the target user.

The source domains refer to the remaining workers with labeled period data in the same dataset. Assuming that there are A source workers, with source worker W_a having I labeled periods, the i-th labeled period from worker W_a is referred to as SP_i^a.

In this study, we assume an activity recognition model based on the U-Net topology consisting of N encoder blocks. As shown in the left part of Figure 3 ($N = 3$), the block of the n-th level consists of multiple layers, with different blocks are represented in different colors in the figure (blue, green,

Algorithm 1 Algorithm for MLKT

Input: J Reference periods from target worker, Source periods from A source workers

1. Block-level Similarity Scoring

Pre-training model on source periods

Extract latent representation from target workers

for $j = 1$ to J **do**
  for $n = 1$ to N **do**
    Extract $F^n(RP_j)$
  end for
end for

Extract latent representation from source workers

for $a = 1$ to A **do**
  for $i = 1$ to I **do**
    Extract $F^n(SP_i^a)$
  end for
end for

Calculate trend similarity

for $n = 1$ to N **do**
  for $a = 1$ to A **do**
    Calculate $\text{SIM}_W^{\text{T}}(W_a, n)$
  end for
end for

2. Multilevel Fine-tuning

for $n = 1$ to N **do**
  for $e = 1$ to N_{ep} **do**
    ASDM selects training periods with similarities
    Train n-th block & ASDM on the selected periods
  end for
end for

and red). For each encoder block, we calculate the *block-level* worker similarity between a target worker and source worker.

Let $F^n(RP_j)$ be the intermediate output (latent representation) for reference period RP_j extracted from the final layer of the n-th encoder block, which is a time series of multivariate feature vectors, each representing a latent vector at each time step of RP_j. We calculate the n-th block-level worker similarity based on $F^n(RP_j)$.

B. Overview

The proposed method, based on adaptive source selection, is divided into two phases as illustrated in Figure 2 and Algorithm 1. The first phase shown in Figure 2 corresponds to the similarity scoring methodology, and the second corresponds to the adaptive fine-tuning strategy. As mentioned in the introduction, because short- and long-term sensor data trends are crucial for complex work recognition, similarities between the target and source workers are calculated based on these trends in the first phase.

In the initial phase, reference periods are used to calculate the trend similarity between the target worker and each of the source workers at each encoding block to find the most suitable data for knowledge transfer by using latent representations at the encoding block. In the final phase, a pre-trained model

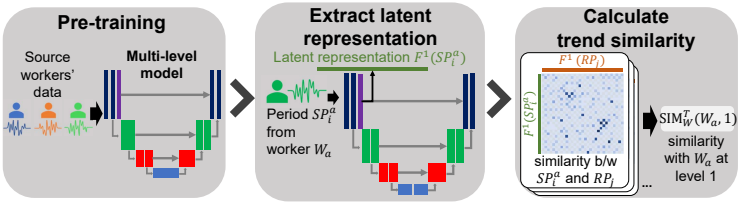

Fig. 3. Block-level trend similarity calculation between source worker W_a periods and the available reference periods from target worker. This is an example of similarity calculation in the first encoding block. Each similarity matrix shows the similarity between a pair of source and target periods. The trend similarity score $\text{SIM}_W^T(\cdot)$ of a block level is the average of the sum of the similarity matrices for the source and target workers calculated by this block. Note: one similarity score is calculated at each block.

is fine-tuned using a level-by-level approach. The proposed ASDM module considers the similarity scores calculated in the first phase and adaptively selects appropriate source workers for each encoding block of the network.

Hence, we propose the multilevel knowledge transfer method, referred to as MLKT. The two phases of the MLKT are described in the following sections.

C. Block-level Similarity Scoring

This section presents the calculation of trend similarity between the target and source workers at each encoding block, considering short- and long-term sensor data trends for the block-level knowledge transfer. An overview of this process is provided in Figure 3. Unlike traditional approaches in which a single similarity or distance metric is calculated between two domains (subjects), the proposed approach calculates the similarity score for each encoder block. Because the objective is to analyze the similarity of a source worker to the target worker at each encoding block, each source worker's period processed by the model is compared with the reference periods processed by the same model at each encoding block.

In the following, we present a similarity calculation method based on the distance between the source and target in terms of the output distributions for each encoding block. However, our method is designed to work with an arbitrary similarity metric.

1) Pre-training Model: The HAR model is pre-trained using all the available data from the source workers. In Figure 3, the U-Net is shown as an example HAR model.

2) Extracting Latent Representation: After model pre-training, the latent representation $F^n(RP_j)$ is extracted for each reference period RP_j from the pre-trained model at the end of each encoding block (n-th block). In addition, the latent representation $F^n(SP_i^a)$ is extracted for each labeled period from each source subject (a-th source subject, i-th period).

3) Calculating Trend Similarity: The block-level trend similarity $\text{SIM}_W^{\text{T}}(W_a, n)$ between source worker W_a and the target worker at the n-th block (level) is calculated by comparing latent representations between the source and target workers. As shown in the rightmost sub-figure of Figure 3, we calculate the similarity between a pair of short segments within the intermediate outputs of target and source periods, forming a similarity matrix. The average value within the matrix is

the similarity between the target reference and labeled source periods at that encoding block. By averaging the similarity of a single source period with all available reference periods, the period-based trend similarity $\mathrm{SIM}_P^{\mathrm{T}}(SP_i^a, n)$ at the n-th block is obtained. Similarly, the calculated average over all labeled periods from source worker W_a constitutes the final worker-based trend similarity score $\mathrm{SIM}_W^{\mathrm{T}}(W_a, n)$ between the target and source worker W_a at the n-th block.

Specifically, $\mathrm{SIM}_W^{\mathrm{T}}(W_a, n)$ is calculated as follows.

$$\mathrm{SIM}_W^{\mathrm{T}}(W_a, n) = \frac{1}{I}\sum_{i=1}^{I} \mathrm{SIM}_P^{\mathrm{T}}(SP_i^a, n), \quad (1)$$

$$\mathrm{SIM}_P^{\mathrm{T}}(SP_i^a, n) = \frac{1}{J}\sum_{j=1}^{J} \mathrm{SIM}^{\mathrm{MMD}}(F^n(SP_i^a), F^n(RP_j)), \quad (2)$$

$$\mathrm{SIM}^{\mathrm{MMD}}(\mathbf{S}, \mathbf{R}) = \frac{1}{N_S N_R}\sum_{k=1}^{N_S}\sum_{l=1}^{N_R} \mathrm{MMD}(\mathcal{H}, \mathbf{S}_k, \mathbf{R}_l), \quad (3)$$

where $\mathbf{S}(\mathbf{R})$ is the intermediate output of an encoder block corresponding to a time series of multivariate vectors, and $\mathbf{S}_k$ is the k-th short segment within $\mathbf{S}$. We assume that $\mathbf{S}$ ($\mathbf{R}$) is divided into N_S (N_R) short segments with the same length. $\mathrm{MMD}(\cdot)$ is a distance metric that calculates the distance between latent representations of source and target workers using the Maximum Mean Discrepancy. The $\mathrm{MMD}(\cdot)$ measures the difference between two segments (distributions), $\mathbf{S}_k$ and $\mathbf{R}_l$, using samples defined in a reproducing kernel Hilbert space (RKHS) as:

$$\mathrm{MMD}(\mathcal{H}, \mathbf{S}_k, \mathbf{R}_l) = \sup_{\|f\|_{\mathcal{H}} \le 1} \left(\mathbb{E}_{x\sim\mathbf{S}_k}[f(x)] - \mathbb{E}_{y\sim\mathbf{R}_l}[f(y)]\right), \quad (4)$$

where $\mathcal{H}$ is the RKHS from a kernel function, and f is indirectly defined by the kernel function. In the equation, the two distributions are embedded into the RKHS, and then the maximum difference in their mean representations is calculated. $\mathbb{E}[\cdot]$ represents an expected value of a given distribution, i.e., the mean, and the upper bound of the difference between the mean representations is yielded by the $\sup(\cdot)$ operation. A kernel such as a Gaussian or polynomial maps data to a high-dimensional space, enabling the capture of complex patterns as in complex packing actions.

The same process is performed to calculate the trend similarity for all source workers at each encoding block. The calculated worker-based trend similarity $\mathrm{SIM}_W^{\mathrm{T}}(W_a, n)$ for each worker and level (a-th worker, n-th block), and period-based trend similarity $\mathrm{SIM}_P^{\mathrm{T}}(SP_i^a, n)$ for each period at each level (n-th block), are used in the multilevel fine-tuning procedure.

D. Multilevel Fine-tuning

This section describes the functionality of the ASDM module and the fine-tuning process of the HAR model. As mentioned in Section I-3, our approach to transferring the knowledge from relevant source workers to an unknown target worker is based on the idea that different levels of the neural network can learn features in different temporal trends, which helps better classify the relevant worker operations depending on the type of data being used to train them. During fine-tuning, particular source workers are selected by the ASDM module at different encoding blocks to improve the overall performance of the network when classifying unknown worker data.

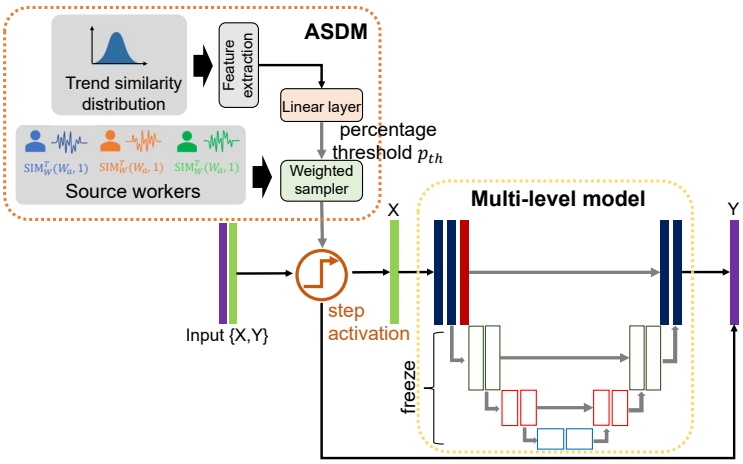

Fig. 4. Diagram of the fine-tuning process for the first encoding block. In this example, only the first blcok is fine-tuned using periods from the source workers selected by the Autonomous Source Discrimination Module (ASDM). Note: Non-colored squares in the HAR model represent convolutional layers not being trained.

The fine-tuning procedure is shown in Figure 4. The ASDM module uses the distribution of trend similarity scores to select appropriate source workers. The step activation layer feeds labeled period data (Input {X, Y} in Figure 4) from only the selected source workers into the HAR model for fine-tuning.

1) Fine-tuning Procedure: For fine-tuning, we adopt the pre-trained model, which is originally trained using all the available data from all the source workers. The fine-tuning process in our method is divided into several stages, with one stage available for each encoder block in the convolutional network. In defining the order of the stages, the shallowest encoding block is the first stage and the deepest encoding block is the last stage. A training stage for the n-th block is composed of the three steps. (i) The ASDM module selects source data that are used to fine-tune the convolutional weights at the n-th block. (ii) We freeze the convolutional weights of all blocks outside the n-th block. (iii) Layers in the n-th block are trained only on the periods selected by the ASDM module. The procedure was iterated for N_{ep} epochs. Note that because the ASDM module is connected to the HAR model and the components in the ASDM module are differentiable, the ASDM module training and HAR model fine-tuning are performed in an end-to-end manner.

The process of freezing and unfreezing the weights is repeated during all fine-tuning stages until all the encoding blocks have been trained.

2) Autonomous Source Discrimination Module (ASDM): In the previous phase, we calculated a set of trend similarities $\mathrm{SIM}_P^{\mathrm{T}}(W_a, \mathbf{S}, n)$ for each level (n-th block). The ASDM module automatically determines a similarity threshold for source worker selection that may maximize the classification

performance during model fine-tuning by referring to the similarity distribution calculated from the similarity set. This is because we believe that the distribution of the similarity scores describes how many similar source workers (periods) are available in the training batch and how similar they are to the target worker.

The ASDM module is composed of two main components, shown in Figure 4: a linear layer and a weighted sampler. Here we explain the source selection procedure for a block of interest in the ASDM module.

(i) We have the trend similarity distribution at each stage (block). We extract statistical values (refer to Table I) from the corresponding distribution and form a distribution feature vector by concatenating the statistical values.

(ii) This vector is fed into the first component of the ASDM module, i.e., the linear layer. The linear layer receives the distribution feature vector and outputs a percentage threshold p_{th}, indicating that the top $p_{th} \times 100$ percent of similar workers' data to the target worker are used in the stage. Note that each source worker is associated with a trend similarity score $\mathrm{SIM}^{\mathrm{T}}_{W}(W_a, n)$. The output p_{th} value varies depending on the prediction error of the previous epoch and internal layer weights.

(iii) The p_{th} value is fed into the weighted sampler. The weighted sampler separates the source workers into those useful and those not useful for fine-tuning the block of interest during the current stage. The weighted sampler feeds the step activation layer with information on which periods should be allowed through.

(iv) The step activation layer receives a binary vector with a size equal to the number of periods in the training batch. For a period selected by the weighted sampler, a value in the vector corresponding to the period is 1; for those not selected, a value corresponding to the period is 0.

(v) Finally, the step activation layer provides period data and labels from the selected periods to the HAR model by referring to the binary vector.

At each epoch, the linear layer is trained according to the procedure. The first stage of fine-tuning is initiated by randomizing the trainable parameters in the linear layer inside the ASDM module. The trainable parameters are then continuously updated based on the model's prediction error, regardless of the stage. As a result, the ASDM module is autonomous in selecting the data used to train at each encoding block.

IV. Evaluation

A. Dataset

The proposed method was evaluated on three logistics datasets: the publicly available Open Packaging (OpenPack) and logistic activity recognition challenge (LARa) datasets [7], [29], which simulate a logistics warehouse environment, and the LOGI dataset, which has data collected from a real-life logistics center.

The LOGI dataset is a private dataset that was collected from four workers at a real logistics center following an instruction manual with sequential operations. Data were collected using a smartwatch (Sony SmartWatch3 SWR50) worn by each worker on the dominant wrist at a sampling rate of 30 Hz. This dataset contains 10 operations (activity classes) to recognize: "Picking," "Replace label," "Assemble box," "Pack in box," "Close box," "Attach label," "Read label," "Put on cart," "Attach address label," "Use pen," and "Unassigned (others)." The dataset has 255 work periods, and the total duration is about 296 minutes.

The OpenPack dataset [7] is a recently published, publicly available dataset containing 58 hours of multimodal data from 16 subjects performing packaging operations in an accurately simulated logistics center workstation. The workers performed packaging operations in four different scenarios. Eleven subjects performed only the first scenario, whereas ten subjects performed the remaining three scenarios. The details for each scenario can be found in [7]. We used accelerometer data collected using ATR TSND151 IMU sensors placed on the subjects' dominant wrists with a sampling rate of 30 Hz. Because the OpenPack dataset is based on an instruction manual, similar to the LOGI dataset, the operations contained are also the same. Scenario 1 adopts a basic setting where workers perform tasks following the prescribed work instructions. Scenarios 2, 3, and 4 assume more realistic situations. In Scenario 2, workers are allowed to change the procedure, and new products are introduced to increase the variety of items. In addition to these changes, Scenario 3 incorporates non-standard actions, such as picking products for multiple orders simultaneously and pre-packing small items (e.g., batteries). Scenario 4 also applies pressure on workers using an alarm sound to replicate a high-demand season, increasing the likelihood of mistakes.

The LARa dataset [29] is a publicly available dataset that contains multiple data types, including accelerometer data from 14 subjects picking products and packaging in three different simulated scenarios. Accelerometer data were captured using MbientLab MMRL IMUs at a sampling rate of 100 Hz. Accelerometer data corresponding to the dominant wrist from six workers who conducted scenarios 2 and 3 for a total of 168 working periods, or 5.6 hours of data was used for the experiment. On average, each period in these scenarios lasts 120 seconds. These scenarios were selected because of their similarity to the actions performed during the traditional packing process in a logistics center. The dataset contained six operations (activity classes) to recognize: "Standing," "Walking," "Cart," "Handling (upward)," "Handling (centered)," and "Handling (downward)." Although the LARa dataset also focuses on packaging works, it is somewhat different from the other datasets because it contains basic action labels (such as standing) as well as complex action labels (such as handling items).

B. Evaluation Methodology

Leave-one-worker-out cross-validation by dataset was used to evaluate the performance of all the tested methods. In the experiment, we assume that each scenario in OpenPack is

TABLE I
EXPERIMENTAL PARAMETERS USED IN THIS STUDY

Parameter	Value
J	2
MMD kernel	Gaussian
# linear layers in ASDM	2
# nodes in 1st layer of ASDM	12
statistical features used in ASDM	Mean, Variance
# epochs (pre-training)	200
# epochs (fine-tuning stage)	100
batch size	64

regarded as a single dataset. The experimental parameters used in the experiment are shown in Table I.

1) Supervised Baselines for Work Activity Recognition:

• **MGA-Net**: The approach presented in [6]. Because we use MGA-Net as a backbone network for our method, we introduce this method in detail. The model structure of MGA-Net is a four-level deep convolutional network based on the topology of U-Net [30], [31] with an added multi-attention head layer at the end of the first convolutional block. The main feature of this method is leveraging sensor data motifs in training the attention layer. This method first identifies frequent sensor data motifs (sensor data segments) for each operation by a motif detection algorithm [6]. Then, the method guides the training of attention heads such that the attention head detects occurrences of each of the motifs. In this experiment, two reference periods with pseudo labels[2] were included along with the source workers' data during training. The same parameters as in [6] were used.

• **LOS-Net**: A CNN-based model for work activity recognition [27] introduced in the related work section. The same parameters as in [27] were used.

2) Source Selection Methods:

• **USSAR/TNNAR**: Unsupervised source selection with a transfer neural network for cross-sensor and cross-user HAR [11]. This is a hybrid method that employs both source selection and network transfer. After selecting Top-K sources by their general and specific domain distances, an adaptive layer was used to reduce the classification discrepancy inside the TNNAR network. Additionally, for a fair comparison, two reference periods with pseudo labels extracted from a Conv-LSTM network similar to TNNAR trained on all source data were added to the selected domain data for transfer. All selection and training parameters in [11] were maintained.

• **ASTTL**: An approach for cross-dataset knowledge transfer [9]. ASTTL focuses on finding the Top-1 most similar domain for transfer by calculating domain similarity using adaptive features in the temporal and spatial spectrums. Given our experimental conditions, we modified the methodology shown in [9] to consider a single worker as an entire domain. With this approach, a single source worker's data is selected for transfer. For a fair comparison, the selected worker's labeled data and two reference periods with pseudo labels, generated after the first training iteration, are used for transfer.

• **MLKT** (ours): This is our proposed method. We used MGA-Net [6] as a backbone. To calculate the similarity between workers for the first convolution layer, which directly processes raw sensor data, we also used the similarity calculated based on raw sensor data, named motif-based similarity. We used the average of the trend and motif-based similarities as our similarity metric. In a nutshell, the motif-based similarity between workers is calculated based on the similarity in motif occurrence timing. As mentioned above, MGA-Net employs several frequent sensor data motifs for each operation in model training. We first employ pseudo operation labels of reference periods to find source workers who share the same motif in the same operation, i.e., identifying if the motif occurs within the same operation for both the target and source workers. We then calculate the average occurrence interval of the motif for each of the found source workers as well as the target worker, and the averaged absolute difference of the interval between the target worker and source worker across all the motifs is used as the inverse of the motif-based similarity. When a motif is not shared by both the target and source workers, the similarity for the motif is zero.

3) Adversarial Training Methods:

• **Contras-GAN**: A contrastive and adversarial learning method for unsupervised domain adaptation [13]. Contras-GAN first employs a bidirectional generative adversarial network to transform samples from the source domain to the target domain and vice versa, and then uses contrastive learning to perform class alignment between the source and target. We adopted the approach for cross-user adaptation. For a fair comparison, the Bi-GAN was trained using all unlabeled periods from the source workers and two reference periods from the target, labeled before performing class-level alignment.

• **AFFAR**: Cross-user domain generalization via domain-specific and domain-invariant adaptive feature fusion learning [20]. AFFAR employs domain-invariant and domain-specific feature extractors to generate three types of loss during generalization. For a fair comparison, an extra source domain containing two reference periods with pseudo-labels, obtained using only the feature extraction and classification modules, was included during generalization.

• **HDCCN**: Transductive transfer learning tuned for convolutional networks presented in [15]. HDCCN aims to preserve feature representation by minimizing the KL divergence between parallel convolutional paths for the target and source domains at each step of convolution. Based on the optimal performance for this method, 100% of unlabeled data from the target domain is used for adaptation, and two periods of reference data with pseudo labels from the original CNN model are used as part of the source data.

4) Unsupervised Domain Generalization Methods:

• **DAPPER**: Label-free performance estimator presented in [19]. DAPPER is divided into two sections, the developer side and the user side. On the development side, existing source

[2]For a fair comparison, reference periods were used in the supervised methods. We first trained the model on source data and then generated pseudo labels of the reference periods using the trained model. After that, we fine-tuned the model on the labeled reference periods and source periods.

TABLE II
RESULTS FOR LEAVE-ONE-WORKER-OUT CROSS-VALIDATION FOR ALL DATASETS.

	LOGI	OpenPack	LARa	AVG
MGA-Net	0.47	0.69	0.68	0.61
LOS-Net	0.41	0.62	0.60	0.54
USSAR/TNNAR	0.26	0.46	0.49	0.40
ASTTL	0.50	0.67	0.70	0.62
Contras-GAN	0.29	0.51	0.67	0.49
AFFAR	0.32	0.59	0.58	0.50
HDCNN	0.36	0.66	0.46	0.49
DAPPER	0.48	0.68	0.67	0.60
DARLING	0.52	0.72	0.70	0.65
FreeMatch	0.40	0.57	0.63	0.53
SSL	0.38	0.58	0.59	0.52
MLKT (ours)	**0.61**	**0.78**	**0.73**	**0.71**

TABLE III
WEIGHTED AVERAGE F1-SCORE FOR ALL METHODS AND WORKERS FROM LOGI DATASET. THESE ARE THE RESULTS OF LEAVE-ONE-WORKER-OUT CROSS-VALIDATION.

Worker	1	2	3	4	AVG
MGA-Net	0.51	0.42	0.47	0.53	0.47
LOS-Net	0.45	0.29	0.42	0.50	0.41
USSAR/TNNAR	0.25	0.21	0.33	0.26	0.26
ASTTL	0.51	0.39	0.57	0.50	0.50
Contras-GAN	0.34	0.25	0.31	0.26	0.29
AFFAR	0.35	0.22	0.36	0.34	0.32
HDCCN	0.37	0.30	0.40	0.35	0.36
DAPPER	0.56	0.32	0.51	0.55	0.48
DARLING	0.63	0.38	0.57	0.52	0.52
FreeMatch	0.43	0.31	0.44	0.42	0.40
SSL	0.44	0.29	0.41	0.40	0.38
MLKT (ours)	**0.68**	**0.45**	**0.66**	**0.64**	**0.61**

domain labeled data is iteratively adapted for diverse virtual scenarios. Then an estimator is trained using the simulated scenarios' adaptation features and accuracies. On the user side, unlabeled data is used to adapt the estimator to a specific target domain. For testing, we employed DAPPER on the "fine-tuning" adaptation method on a backbone consisting of CNN layers and fully connected layers. For fine-tuning adaptation, we use two reference periods with pseudo labels obtained from a pre-trained model using all source workers' data.

• **DARLING**: Unsupervised domain generalization by performing domain-aware representation learning [18]. DARLING focuses on generalizing models to unseen domains by first performing domain-irrelevant unsupervised learning followed by domain-specific negative sample generation to perform contrastive learning. For testing, we exchange the idea from object domains to workers. Furthermore, the RESNET18 architecture is modified to receive a 3-channel one-dimensional input corresponding to a section of the work period. We also include two reference periods with pseudo labels obtained after pre-training for fine-tuning.

• **FreeMatch**: A method for self-adaptive thresholding during semi-supervised learning [32]. A similar approach to our own applied to image classification. FreeMatch consists of two stages: Self-Adapting Thresholding (SAT), to adapt the model training according to its learning status by using the exponential moving average (EMA) of unlabeled data confidence, and Self-Adaptive class Fairness regularization (SAF) to encourage diverse predictions during the early stages of training.

5) Self-supervised Learning:

• **SSL**: We evaluated a model consisting of a publicly available encoder (feature extractor) pre-trained on 700,000 days of unlabeled data via self-supervised learning [33]. We trained an output layer using source workers' data and two reference periods with pseudo labels.

C. Results

1) Recognition Accuracy: Table II shows the weighted average F1-score obtained by each method across all three logistic datasets. Note that the results for OpenPack are the averages over the four scenarios. Being the first transfer learning method designed for complex activity recognition, MLKT outperforms all other baselines in the three datasets. Overall, it can be said that the proposed method provides a classification accuracy improvement of 8 to 20% compared to existing state-of-the-art methods. Surprisingly, it was the adversarial domain adaptation/generalization methods (Contras-GAN, AFFAR, and HDCCN) that had the hardest time with this particular task. These results indicate the difficulty in acquiring domain-invariant features from the complex packaging works with short- and long-term trends through adversarial learning with limited sensor data from target workers. As for the source selection methods, ASTTL showed F1-score values comparable to those of MGA-Net, proving the importance of transferring based on temporal and spatial similarities across workers. The unsupervised domain generalization methods had a better performance than other baselines, with DARLING obtaining the second-highest F1-score across all the datasets. Finally, the performance of the self-supervised learning (SSL) method based on the publicly available encoder was very poor. This may be because it was pre-trained on simple daily life data. In addition, because the SSL model is already trained with a 10-second time window, it may not capture any shorter or longer period of trend.

Also, in Table II, it can be observed how the performance of all methods varies significantly between datasets. In particular, all methods struggled to transfer adequately in the LOGI dataset. Detailed results of the LOGI dataset can be seen in Table III. The F1-score for MLKT was higher than other baselines across all workers. It can be noted that all methods have trouble transferring to Worker 2, which may be due to a gap in experience compared to the other workers. Workers 1, 3, and 4 have over a decade of experience in packing operations, and Worker 2 has less than six months.

Table IV shows the performance of all the methods across the four different scenarios from the OpenPack dataset. Our method obtained a weighted average F1-score of 0.78 overall for this dataset, while the second best-performing baseline, DARLING, scored 0.72. Similar to the case in LOGI, our method achieved a 5-8% improvement over the second-best in all four scenarios. The F1-score of LOS-Net, which is a supervised baseline, for Scenario 1 was 0.62, which is poorer than that reported in [7] because the test in [7] predicted

TABLE IV
RESULTS FOR LEAVE-ONE-WORKER-OUT CROSS-VALIDATION FOR ALL SCENARIOS IN THE OPENPACK DATASET.

Scenario	1	2	3	4	AVG
MGA-Net	0.76	0.74	0.66	0.61	0.69
LOS-Net	0.67	0.64	0.63	0.54	0.62
USSAR/TNNAR	0.50	0.47	0.45	0.42	0.46
ASTTL	0.67	0.70	0.69	0.63	0.67
Contras-GAN	0.55	0.52	0.50	0.46	0.51
AFFAR	0.64	0.61	0.58	0.54	0.59
HDCNN	0.69	0.69	0.65	0.60	0.66
DAPPER	0.73	0.71	0.68	0.61	0.68
DARLING	0.75	0.74	0.71	0.66	0.72
FreeMatch	0.62	0.58	0.56	0.52	0.57
SSL	0.64	0.61	0.55	0.52	0.58
MLKT (ours)	**0.82**	**0.81**	**0.77**	**0.71**	**0.78**

TABLE V
WEIGHTED AVERAGE F1-SCORE FOR ALL METHODS AND WORKERS FROM LARA DATASET. THESE ARE THE RESULTS OF LEAVE-ONE-WORKER-OUT CROSS-VALIDATION.

Worker	7	8	9	10	13	14	AVG
MGA-Net	0.71	0.68	0.68	0.65	0.72	0.66	0.68
LOS-Net	0.68	0.57	0.55	0.56	0.59	0.61	0.60
USSAR/TNNAR	0.51	0.50	0.49	0.39	0.54	0.50	0.49
ASTTL	0.70	0.63	0.69	**0.75**	0.71	0.71	0.70
Contras-GAN	0.67	0.68	0.59	0.70	0.74	0.64	0.67
AFFAR	0.52	0.62	0.61	0.6	0.58	0.55	0.58
HDCCN	0.40	0.51	0.45	0.47	0.44	0.46	0.46
DAPPER	0.67	0.70	0.62	0.62	0.75	0.66	0.67
DARLING	0.68	**0.75**	0.65	0.67	0.76	**0.73**	0.70
FreeMatch	0.64	0.67	0.61	0.60	0.69	0.61	0.63
SSL	0.53	0.62	0.60	0.63	0.60	0.58	0.59
MLKT (ours)	**0.75**	0.74	**0.71**	0.69	**0.78**	0.71	**0.73**

a class label for each 1-sec data window whereas our test predicted a fine-grained label for each data point. Observing Table IV, it appears that the classification performance for all the methods trends downwards from the first to the fourth scenario. Given the implied increased task complexity for Scenarios 3 and 4 [7], a slight decrease in model performance was expected. Interestingly, the methods most affected by this were the supervised models, MGA-Net and LOS-Net, with their weighted average F1-score falling by more than 15%. This may be due to a lack of adaptability in these models or the high data requirements needed to train them effectively.

Table V shows the results for the LARa dataset. Once again, MLKT outperforms state-of-the-art baselines overall, achieving a weighted average F1-score of 0.73 for all workers combined. On this dataset, however, the improvement of our method over existing baselines was limited to less than 5%. Table V also highlights that both ASTTL and DARLING slightly outperformed MLKT on three occasions. We believe this could be due to the limited number and complexity of operation classes available in the LARa dataset. This dataset shows the smallest deviation in performance across the methods.

2) Ablation Study: Additionally, we evaluate the impact of the different components of MLKT. We introduce the following methods for comparison.

• **MGA-Net**: The baseline of this study as introduced in [6].

• **Top-1 (trend-only)**: A variant of MLKT that implements the Top-1 source selection strategy for transfer. The Top-1 source worker is selected based solely on their trend-based worker similarity.

• **Top-1**: A variant of MLKT that implements the Top-1 source selection strategy for transfer. The average of trend-based and motif-based similarities is considered for selection for the first layer.

• **W/o ASDM (trend-only)**: A variant of MLKT where the data percentage threshold for transfer is determined manually. For adequate comparison, we chose a data percentage threshold of 70%, since it shows the best performance across all datasets in our preliminary experiment. Source workers are ranked using their trend-based worker similarity only. Top-70% most similar source workers are used during fine-tuning.

• **W/o ASDM**: This variant is similar to the previous one. However, workers are ranked by averaging the trend-based and motif-based similarity scores.

• **W/ASDM (trend-only)**: A variant of MLKT where the ASDM module adaptively determines the data percentage threshold using solely the trend-similarity distribution.

TABLE VI
RESULTS OF ABLATION STUDY ACROSS ALL DATASETS.

	LOGI	OpenPack	LARa	AVG
MGA-Net	0.47	0.69	0.68	0.61
Top-1 (trend-only)	0.48	0.69	0.68	0.62
Top-1	0.50	0.72	0.70	0.64
W/o ASDM (trend-only)	0.54	0.66	0.71	0.63
W/o ASDM	0.57	0.68	0.71	0.65
W/ASDM (trend-only)	0.58	0.73	0.69	0.66
MLKT	0.61	0.78	0.74	0.71

Here we evaluate the Top-1 methods because ASTTL, which is the best source selection baseline, also uses the top-1 source worker. We use the percentage threshold of 70% in the W/o ASDM methods because the LOGI dataset has only three source workers. 70% corresponds to a case where two source workers are used. In MGA-Net, all three source workers are used.

Table VI shows the results of the ablation study across all the datasets. By comparing the results between Top-1 and Top-1 (trend-only), we can observe improvements of about 2-3% by incorporating the motif-based similarity when recognizing complex activities.

The superiority of W/o ASDM and Top-1 varied depending on the datasets, with W/o ASDM variants performing worse in the OpenPack dataset compared to LOGI and LARa. This could be due to the number of workers available in each dataset. While the data percentage threshold was manually set at 70%, the actual number of workers selected varied by dataset, with LOGI having a threshold of 2 workers, OpenPack having 6-7 workers, and LARa having 5 workers. While having more workers can provide the model with more general information about the overall task, it can also result in negative transfer, as observed in the OpenPack dataset.

Finally, we can see the impact of the ASDM module adaptively changing the data threshold during training in W/ASDM (trend-only) and MLKT. W/ASDM (trend-only) performed

TABLE VII
DETAILED RESULTS OF ABLATION STUDY FOR THE LOGI DATASET.

Worker	1	2	3	4	AVG
MGA-Net	0.51	0.42	0.47	0.53	0.47
Top-1 (trend-only)	0.53	0.40	0.50	0.51	0.48
Top-1	0.51	0.39	0.57	0.50	0.50
W/o ASDM (trend-only)	0.56	0.44	0.60	0.58	0.54
W/o ASDM	0.67	0.43	0.60	0.59	0.57
W/ASDM (trend-only)	0.63	0.44	0.64	0.63	0.58
MLKT	**0.68**	**0.45**	**0.66**	**0.64**	**0.61**

TABLE VIII
DETAILED RESULTS OF ABLATION STUDY FOR THE LARA DATASET.

Worker	7	8	9	10	13	14	AVG
MGA-Net	0.71	0.68	0.68	0.65	0.72	0.66	0.68
Top-1 (trend-only)	0.70	0.63	0.70	0.64	0.72	0.66	0.68
Top-1	0.70	0.63	0.69	**0.75**	0.71	**0.71**	0.70
W/o ASDM (trend-only)	0.72	**0.78**	0.69	0.68	0.71	0.70	0.71
W/o ASDM	0.73	0.75	0.68	0.70	0.73	0.70	0.71
W/ASDM (trend-only)	0.71	0.74	0.69	0.64	0.69	0.68	0.69
MLKT	**0.75**	0.74	**0.71**	0.69	**0.78**	**0.71**	**0.73**

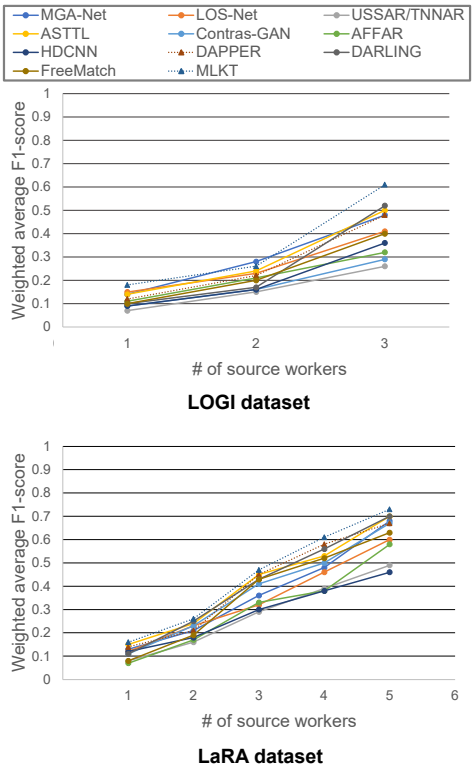

Fig. 5. Results for work activity recognition with a varied number of source workers' data for training. These are the average results for the LOGI and LARa datasets.

better than W/o ASDM (trend-only) and Top-1 (trend-only) in the LOGI and OpenPack datasets. However, it was surpassed by W/o ASDM (trend-only) in the LARa dataset. This may be because the trend-based similarity distribution values are very compact, given that the data complexity of LARa is limited compared to that of LOGI and OpenPack. As a result, the ASDM module may have struggled to find the optimal data percentage threshold during training. On the other hand, MLKT outperforms all other variants. This highlights the importance of adapting the data threshold during training and incorporating motifs to calculate worker similarity for improved knowledge transfer.

Table VII and Table VIII show the average F1-score values of each target worker obtained by all model variations for the LOGI and LARa datasets, respectively. As shown in the results Top-1 could not outperform MGA-Net in all cases even though the Top-1 method employs training data from the most similar source worker for each encoding block. For example, in the LOGI dataset, the performance for Top-1 is poorer than that of MGA-Net in two workers. In addition, in the LARa dataset, the performance for Top-1 is poorer than that of MGA-Net in three workers. In contrast, MLKT always outperformed MGA-Net in all cases and Top-1 in most cases.

These results suggest that just selecting similar source workers for each block does not always improve the recognition performance in complex work activity recognition, and the combination of block-level source selection and adaptive source selection by ASDM is crucial.

V. DISCUSSION

A. Number of Source Workers

Here, we discuss the effect of limiting the number of available source workers for knowledge transfer. For this test, we initially selected a single worker from the dataset as the target and then randomly selected the specified number of source workers from the remaining workers in the dataset. We repeated this process until all workers in the dataset had been used as targets.

Figure 5 shows the results for the LOGI and LARa datasets. In both datasets, the average performance of MLKT was the best in most tests. For the LOGI dataset, both MLKT and DARLING show a marked improvement when increasing the number of source workers from two to three, with DARLING moving from being the eighth method overall to the second. As expected, given the very limited number of workers in the LOGI dataset, the results for most methods show minimal improvement between one and two sources. In contrast, in the LARa dataset, most methods remain in relatively the same positions across all tests. In this dataset, MLKT outperforms other baselines overall, with more limited improvement as the number of source workers increases. We believe this could be due to the low dispersion of workers' data, resulting in generally better performance by transferring trend and motif features from any available worker, but reducing the impact as more similarly performing source workers are added.

Figure 6 shows the separate results for each OpenPack scenario. Unlike the results for LOGI and LARa, the results

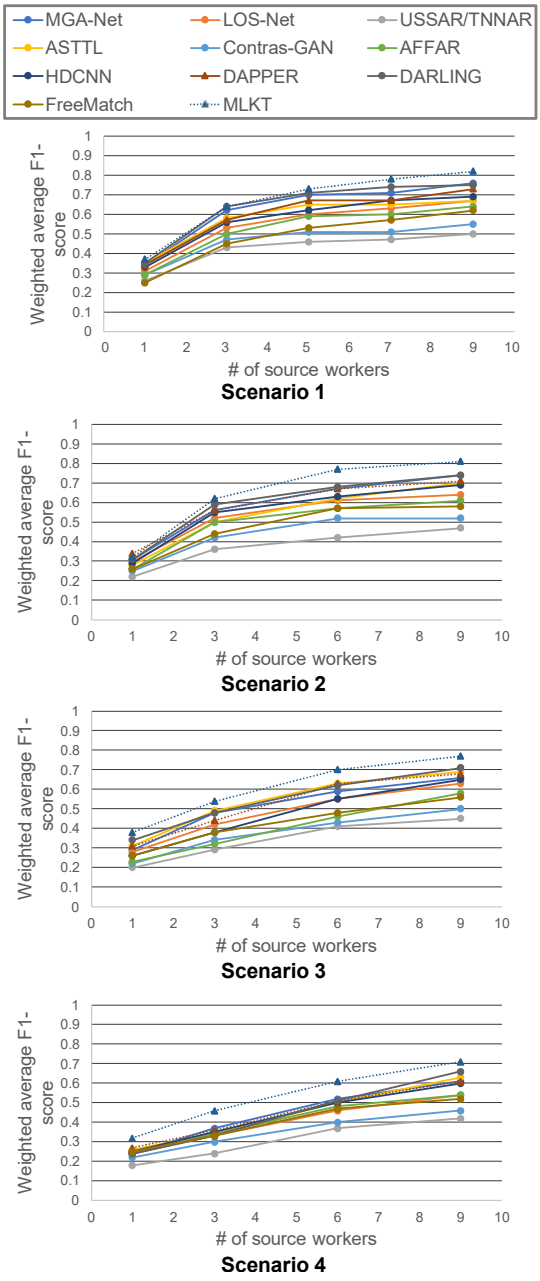

Fig. 6. Results for work activity recognition with a varied number of source workers' data for training. These are the average results for each of the four scenarios from the OpenPack dataset.

in Scenario 1 and Scenario 2 of OpenPack do not show significant changes in performance as the number of source workers increases. In these scenarios, we note that MLKT does not offer a great improvement compared to the baselines when very limited options are available for transfer. In particular, when only one source worker is available, the performance of MLKT is worse than that of state-of-the-art methods. However, its improvement over the baselines as the number of sources increases is notable, especially in Scenario 2. This may be because Scenario 2 simulates irregular operation patterns, such as scanning items inside or outside the box, placing the shipping label on the side of the box that the worker finds more comfortable, etc. These alterations increase the variety of short- and long-term data between workers but also increase the importance of finding workers with similar trends.

In Scenario 3 and Scenario 4, MLKT outperforms all other baselines even with a limited number of source workers. We believe this is due to the increasing irregularity in operations introduced in Scenarios 3 and 4. The complexity of these scenarios likely benefits from transferring both short- and long-term features.

B. Computation Costs

Compared to state-of-the-art MGA-Net, MLKT requires an additional similarity score calculation process before neural network training. We measure the computation times related to the similarity score calculation by employing the LOGI dataset. Before calculating the similarity scores, we should acquire latent representations of two reference periods from a target worker. The computation time for that process was 0.11 seconds when we used an NVIDIA Tesla T4 GPU. After that, we calculate a similarity matrix for each pair of reference and source period for each encoding block. The total computation times for the 1st, second, and third blocks are 427.0 s, 269.7 s, and 191.8 s, respectively. As above, it takes about 800 seconds to calculate similarity scores for each target worker. However, the inference time of MLKT is identical to that of MGA-Net because both utilize the same network structure.

VI. Conclusion

We presented a novel method for knowledge transfer in complex activity recognition for logistics scenarios. We introduced a multilevel fine-tuning approach that adaptively selects, during training, the most relevant source data at each level of network encoding. We evaluated our method against 11 state-of-the-art HAR methods and demonstrated its efficiency using three separate logistic operation datasets.

As a part of our future work, we plan to apply our method to other complex activities such as cooking activities. Note that, compared to cooking activities, work activities do not contain repetitive motions in many cases, making it difficult to recognize these activities with their periodic patterns. Therefore, identifying short atomic actions (short-term patterns) and capturing temporal orders of them (long-term patterns) is more crucial in work activity recognition. In addition, reducing annotation costs regarding source worker data is one of our important future directions. By employing a self-supervised learning scheme, we plan to pre-train a HAR network on unlabeled data from source workers, and then fine-tune it with limited labeled data from a target worker.

Acknowledgement

This study is partially supported by JSPS KAKENHI Grant Number JP21H05299, Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things(No.2023B1212010007), China NSFC Grant(U2001207, 62472366), the Project of DEGP (No.2024GCZX003, 2023KCXTD042, 2021ZDZX1068).

REFERENCES

[1] M. Consult. (2022) The state of retail & e-commerce h2 2022 report. [Online]. Available: https://go.morningconsult.com/State-of-Retail-and-E-Commerce-Report-Download.html

[2] J. Alcedo, A. Cavallo, B. Dwyer, P. Mishra, and A. Spilimbergo, "E-commerce during covid: Stylized facts from 47 economies," National Bureau of Economic Research, Working Paper 29729, February 2022. [Online]. Available: http://www.nber.org/papers/w29729

[3] M. Klumpp, M. Hesenius, O. Meyer, C. Ruiner, and V. Gruhn, "Production logistics and human-computer interaction—state-of-the-art, challenges and requirements for the future," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3691–3709, 2019.

[4] C. Reining, F. Niemann, F. Moya Rueda, G. A. Fink, and M. ten Hompel, "Human activity recognition for production and logistics—a systematic literature review," *Information*, vol. 10, no. 8, p. 245, 2019.

[5] V. Yavas and Y. D. Ozkan-Ozen, "Logistics centers in the new industrial era: A proposed framework for logistics center 4.0," *Transportation Research Part E: Logistics and Transportation Review*, vol. 135, mar 2020.

[6] J. Morales, N. Yoshimura, Q. Xia, A. Wada, Y. Namioka, and T. Maekawa, "Acceleration-based human activity recognition of packaging tasks using motif-guided attention networks," in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022, pp. 1–12.

[7] N. Yoshimura, J. Morales, T. Maekawa, and T. Hara, "Openpack: A large-scale dataset for recognizing packaging works in iot-enabled logistic environments," in *2024 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Los Alamitos, CA, USA: IEEE Computer Society, mar 2024, pp. 90–97. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/PerCom59722.2024.10494448

[8] N. Hernandez, J. Lundström, J. Favela, I. Mcchesney, and B. Arnrich, "Literature Review on Transfer Learning for Human Activity Recognition Using Mobile and Wearable Devices with Environmental Technology," *SN Computer Science*, vol. 1, no. 2, pp. 1–16, 2020. [Online]. Available: https://doi.org/10.1007/s42979-020-0070-4

[9] X. Qin, Y. Chen, J. Wang, and C. Yu, "Cross-dataset activity recognition via adaptive spatial-temporal transfer learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, 2019.

[10] M. J. Afridi, A. Ross, and E. M. Shapiro, "On automated source selection for transfer learning in convolutional neural networks," *Pattern Recognition*, vol. 73, pp. 65–75, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317302881

[11] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," *arXiv preprint:1807.07963*, 2018.

[12] Q. Chen, Y. Liu, Z. Wang, I. Wassell, and K. Chetty, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7976–7985.

[13] A. R. Sanabria, F. Zambonelli, S. Dobson, and J. Ye, "ContrasGAN: Unsupervised domain adaptation in Human Activity Recognition via adversarial and contrastive learning," *Pervasive and Mobile Computing*, vol. 78, p. 101477, 2021. [Online]. Available: https://doi.org/10.1016/j.pmcj.2021.101477

[14] S. Suh, V. F. Rey, and P. Lukowicz, "Adversarial Deep Feature Extraction Network for User Independent Human Activity Recognition," *2022 IEEE International Conference on Pervasive Computing and Communications, PerCom 2022*, pp. 217–226, 2022.

[15] M. A. A. H. Khan, N. Roy, and A. Misra, "Scaling Human Activity Recognition via Deep Learning-based Domain Adaptation," *2018 IEEE International Conference on Pervasive Computing and Communications, PerCom 2018*, pp. 1–9, 2018.

[16] H. Niu, H. Q. Ung, and S. Wada, "Source domain selection for cross-house human activity recognition with ambient sensors," in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022, pp. 754–759.

[17] S. Fan, Y. Jia, and C. Jia, "A feature selection and classification method for activity recognition based on an inertial sensing unit," *Information*, vol. 10, no. 10, 2019. [Online]. Available: https://www.mdpi.com/2078-2489/10/10/290

[18] X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu, "Towards unsupervised domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4910–4920.

[19] T. Gong, Y. Kim, A. Orzikulova, Y. Liu, S. J. Hwang, J. Shin, and S.-J. Lee, "Dapper: Label-free performance estimation after personalization for heterogeneous mobile sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, no. 2, jun 2023. [Online]. Available: https://doi.org/10.1145/3596256

[20] X. Qin, J. Wang, Y. Chen, W. Lu, and X. Jiang, "Domain generalization for activity recognition via adaptive feature fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, pp. 1 – 21, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251018355

[21] A. Hoelzemann and K. Van Laerhoven, "Digging deeper: towards a better understanding of transfer learning for human activity recognition," in *Proceedings of the 2020 ACM International Symposium on Wearable Computers*, ser. ISWC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 50–54. [Online]. Available: https://doi.org/10.1145/3410531.3414311

[22] Q. Xia, A. Wada, J. Korpela, T. Maekawa, and Y. Namioka, "Unsupervised factory activity recognition with wearable sensors using process instruction information," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 2, p. 60, 2019.

[23] Q. Xia, J. Korpela, Y. Namioka, and T. Maekawa, "Robust Unsupervised Factory Activity Recognition with Body-worn Accelerometer Using Temporal Structure of Multiple Sensor Data Motifs," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–30, sep 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3411836

[24] T. Maekawa, D. Nakai, K. Ohara, and Y. Namioka, "Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory," in *UbiComp 2016*, 2016, pp. 1088–1099.

[25] W. Tao, Z. H. Lai, M. C. Leu, and Z. Yin, "Worker Activity Recognition in Smart Manufacturing Using IMU and sEMG Signals with Convolutional Neural Networks," *Procedia Manufacturing*, vol. 26, pp. 1159–1166, jan 2018.

[26] A. S. Syed, Z. S. Syed, and A. K. Memon, "Continuous human activity recognition in logistics from inertial sensor data using temporal convolutions in CNN," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 10, pp. 597–602, 2020.

[27] N. Yoshimura, T. Maekawa, T. Hara, A. Wada, and Y. Namioka, "Acceleration-based activity recognition of repetitive works with lightweight ordered-work segmentation network," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, 2022.

[28] M. Gjoreski, S. Kalabakov, M. Luštrek, M. Gams, and H. Gjoreski, "Cross-dataset deep transfer learning for activity recognition," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC '19 Adjunct. New York, NY, USA: Association for Computing Machinery, 2019, p. 714–718. [Online]. Available: https://doi.org/10.1145/3341162.3344865

[29] F. Niemann, C. Reining, F. Moya Rueda, N. R. Nair, J. A. Steffens, G. A. Fink, and M. Ten Hompel, "Lara: Creating a dataset for human activity recognition in logistics using semantic attributes," *Sensors*, vol. 20, no. 15, p. 4083, 2020.

[30] Y. Zhang, Y. Zhang, Z. Zhang, J. Bao, and Y. Song, "Human activity recognition based on time series analysis using u-net," *arXiv preprint arXiv:1809.08113*, 2018.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint:1505.04597*, 2015.

[32] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie, "Freematch: Self-adaptive thresholding for semi-supervised learning," *arXiv preprint:2205.07246*, 2023.

[33] H. Yuan, S. Chan, A. P. Creagh, C. Tong, A. Acquah, D. A. Clifton, and A. Doherty, "Self-supervised learning for human activity recognition using 700,000 person-days of wearable data," *npj Digital Medicine*, vol. 7, no. 1, Apr. 2024. [Online]. Available: http://dx.doi.org/10.1038/s41746-024-01062-3